SAME ANSWERS, DIFFERENT TARGETS: THE HIDDEN JOINT LAW OF STRUCTURED SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Teaching the same answers together can produce different training targets. Two Python workers must return complementary subsets of an input, but the task does not require them to share an implementation style. We construct datasets with identical local programs, marginals, relations, and legal support, yet different frequencies of complete responses: their *joint laws*. Across three paired Qwen3-8B runs, this change moves held-out same-style generation from 24.3–25.1% to 72.0–72.6%; a common-update-denominator rerun reproduces the contrast at one existing seed. Changing only the weights on the same $1{,}024$ scored validation responses also reverses likelihood-based checkpoint selection on all 32 held-out specifications. A classical minimum-information projection separates task-required from constructor-added dependence. On released programs, upweighting same-source pairings raises a prespecified voter's wrong acceptance by 4.31 points over 75 tasks with unseen inputs. A separate experiment holds the actual first program fixed: a new second attempt shares more errors and supplies less rescue, reducing at-least-one success by about one point. The joint law is thus a learner-visible part of supervision with consequences for both model selection and what one answer adds to another.

1 INTRODUCTION

Grouped supervision is often audited one output at a time: each label is valid, each marginal is as intended, and every declared relation is satisfied. But when related outputs share one autoregressive response, these checks leave open how often individually legal outputs occur together. Those frequencies are part of the conditional target a maximum-likelihood learner fits. A data constructor therefore chooses not only which answers to teach, but which answers become evidence about one another.

Our two-worker task makes the choice concrete. One response asks two Python workers to divide an integer list into complementary residue subsets. The task specifies what each computes and how the pair covers the input, but not whether both use a comprehension, both an explicit loop, or different valid styles. Two datasets contain the same correct programs, worker marginals, legal pairs, and complementarity relation, yet assign different frequencies to complete responses. Split them into worker-local examples and the training multisets are identical. Keep the workers together and the learner receives different conditional targets.

Across three paired Qwen3-8B runs, changing only legal-response multiplicities moves held-out same-style behavior from 24.3–25.1% to 72.0–72.6%. A common-update-denominator rerun reproduces the contrast at one seed: 47.75 points (95% interval $[45.36, 50.10]$).

The same omission reaches a routine decision: likelihood-based checkpoint selection averages scores over a validation law, and worker-local audits do not specify its complete-response weights. Holding both checkpoints and the same $1{,}024$ scored legal responses fixed, changing only those weights reverses their ranking on all 32 held-out specifications in every tested pair. Section 5.2 asks how much additional information suffices to determine this choice.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We then follow the variable into use. On released programs, a prespecified voting consumer changes its decisions on inputs absent from the public examples. A separate continuation test holds the actual first program fixed and asks what a newly generated second attempt adds: more shared errors on individual inputs, but less rescue of a first program that fails somewhere on the task. Other executable and released-program carriers show the claim's boundary—assembly changes behavior and composition without guaranteeing a correctness gain.

Prior work establishes useful coupling, consequential packing, and diverse or coordinated generation (De Mijolla et al., 2025; Ravfogel et al., 2025; Krell et al., 2021; Wang et al., 2025; Dong et al., 2025; Li et al., 2025; Zhao et al., 2026; Handa et al., 2026; Li et al., 2026). Our question begins after candidate answers and relations are fixed: can reassembling identical same-task atoms change learning when local supervision is unchanged? The interface is real: same-question mathematical SFT already serializes three solutions into one response (Lee et al., 2026), although that comparison regenerates the answers rather than fixing their inventory.

We make this residual choice auditable with *minimum-sufficient coupling* (Minimum), the classical KL projection of independent marginals onto the laws that satisfy the declared relations. Its diagnostic role is to separate dependence required by the task specification from additional dependence introduced by data construction.

Our contributions are twofold:

- **A constructor-controlled grouped target.** Matched local answers, marginals, support, and relations leave complete-response multiplicity undetermined. A classical minimum-information projection separates specification-required from constructor-added dependence.
- **Learner, selection, and decision consequences.** Matched-content interventions change held-out cross-output behavior, reverse likelihood selection on identical scored rows, and change a prespecified program voter. A fixed-first-program test finds more shared errors and less rescue from a new second attempt.

2 Structured Supervision Is a Joint-Law Design Problem

A supervision specification constrains local outputs and their relations. We ask which distributions over complete groups—their *joint laws*— remain compatible with that specification.

Let a group contain m output nodes $Y = (Y_1, \dots, Y_m)$, with $Y_i \in \mathcal{Y}_i$. The supervision specification freezes a candidate marginal p_i at every node and a collection of *typed relations*. A relation e has an explicit semantics, a computable score $r_e(Y_e)$ on the nodes it connects, and a required level τ_e. Let $\Omega \subseteq \prod_i \mathcal{Y}_i$ be the declared legal support; hard legality is encoded by membership in Ω. The feasible joint laws are

$$\mathcal{F} = \{\pi : \mathrm{supp}(\pi) \subseteq \Omega, \quad \pi_i = p_i \ \forall i, \quad \mathbb{E}_\pi[r_e(Y_e)] \geq \tau_e \ \forall e\}\,. \tag{1}$$

An executable relation can equivalently be folded into Ω when it is a hard legality condition; the moment constraints above are used for graded or soft requirements. The product law $q(y) = \prod_i p_i(y_i)$ preserves every node marginal by construction, but it need not be feasible when the task requires coordination. Throughout this paper, the candidate sets, marginals, relations, and thresholds are declared inputs; learning or repairing them is a different problem.

In the worker example above, the two outputs must choose complementary residue subsets and may use either valid implementation style. Complementary orientation is *task-required*; cross-worker style matching is not.

For the smallest construction, let orientation and style be binary and uniform at each worker. Choose the two complementary orientations equally and, writing Z_i for worker i's style, couple the styles symmetrically so that $\Pr(Z_1 = Z_2) = (1 + \rho)/2$, $0 \leq \rho \leq 1$. Every ρ preserves both worker marginals and relation fidelity; every $0 < \rho < 1$ also retains all four style pairs. Thus fixed marginals, exact relation satisfaction, and full legal support still do not identify the supervision distribution. Section 4.1 realizes this residual frequency freedom as an exact-prompt training intervention, and Section 4.4 varies its strength at four discrete levels.

Orientation is a *task-required decision*; style is a *task-free choice*, meaning only that the declared specification does not require it to co-vary.

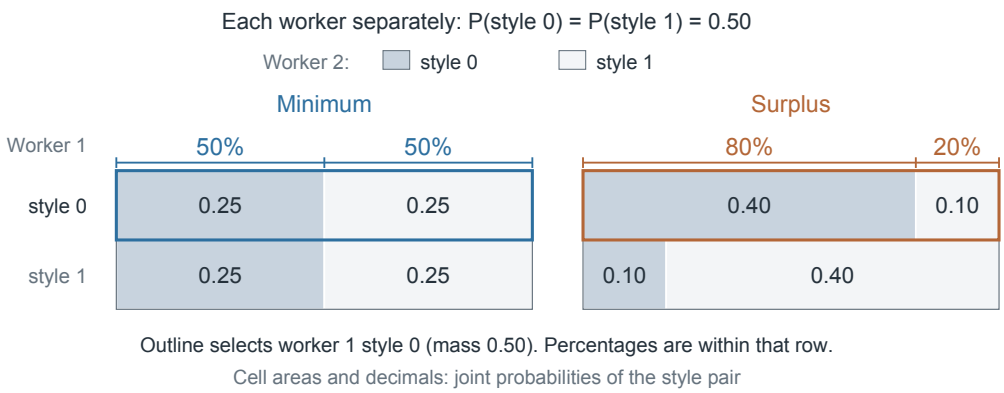


Figure 1: **Local equality need not identify a grouped target.** Cell areas show exact style-pair probabilities. The outlined row selects worker-1 style 0 (mass 0.50); its internal width split gives worker 2's conditional target. Complementary orientation is unchanged and omitted.

Total correlation $\mathrm{TC}(\pi) = \sum_i H_\pi(Y_i) - H_\pi(Y)$ measures their aggregate dependence in bits. Here complementary orientation costs one bit. The $\rho = 0$ feasible law pays only that bit, while $\rho = 1$ pays another bit by tying styles. The product law is under-coupled because it satisfies complementarity only half the time; the latter law is over-coupled because it also coordinates the task-free choice. Relation audits detect the first but not the second.

Figure 1 makes the residual degree of freedom and its learner-visibility boundary explicit.

When is the joint law visible to the learner? The downstream channel depends on the supervision interface. If each node is scored separately and the population objective decomposes over nodes, then any two laws π and π' with the same marginals induce the same expected objective:

$$\mathbb{E}_\pi\left[\sum_i \ell_{i,\theta}(Y_i)\right] = \sum_i \mathbb{E}_{p_i}[\ell_{i,\theta}(Y_i)] = \mathbb{E}_{\pi'}\left[\sum_i \ell_{i,\theta}(Y_i)\right]. \tag{2}$$

Several answers to one prompt trained under separate per-response losses have this invariant population objective. Their finite-sample information can still differ: one shared task-free choice across m nodes turns Nm valid labels into N independent draws. Appendix C.1 quantifies this efficiency channel. We focus instead on related outputs sharing one autoregressive response, where pairing changes the fitted conditional target.

Call two laws *audit-equivalent* when they share node marginals, relation scores, and legal support. Fix a learner-visible context x and an injective serialization s_x of legal tuples. Audit-equivalent laws induce the same node-separable objective by Eq. (2), but grouped log loss has distinct Bayes targets. With $R_{\pi,x}(g) = \mathbb{E}_\pi[-\log_2 g(s_x(Y) \mid x)]$, any two audit-equivalent laws $\pi \neq \pi'$ with common support give

$$R_{\pi,x}((s_x)_\#\pi') - R_{\pi,x}((s_x)_\#\pi) = D_{\mathrm{KL}}((s_x)_\#\pi \| (s_x)_\#\pi') = D_{\mathrm{KL}}(\pi\|\pi') > 0. \tag{3}$$

Autoregressive token losses through EOS sum to complete-sequence log loss: local-and-support equivalence does not imply grouped-target equivalence. Complete rows encode their empirical multiplicities; local summaries do not. When two laws share the first-slot marginal, the chain rule localizes their discrepancy to $\mathbb{E}_{Y_1} D_{\mathrm{KL}}(\pi(Y_2 \mid Y_1)\|\pi'(Y_2 \mid Y_1))$. This motivates the fixed-first-program readout in Section 6.2, which measures conditional behavior rather than this KL. The identity concerns population targets; whether finite learners under token-normalized SGD track the injected pairing structure is the empirical question addressed next.

3 Minimum-Sufficient Coupling: A Specification-Relative Reference

We make the residual freedom of Section 2 auditable by taking the product law as a zero-dependence reference and departing from it only as far as the specification requires.

Minimum-sufficient coupling. Let $q(y) = \prod_i p_i(y_i)$ and let $\mathcal{F}$ be the feasible set in Eq. (1). The classical information projection (Csiszár, 1975) defines

$$\pi_{\min} = \arg\min_{\pi \in \mathcal{F}} D_{\mathrm{KL}}(\pi \,\|\, q)\,. \tag{4}$$

Every $\pi \in \mathcal{F}$ has the frozen marginals p_i, so

$$D_{\mathrm{KL}}(\pi \| q) = \sum_i H(p_i) - H(\pi) = \mathrm{TC}(\pi). \tag{5}$$

Minimum has least total correlation and largest joint entropy among feasible laws. It is a diagnostic reference, not a prescription for the best training law. The projection is specification-relative: changing the output partition, conditioning variables, marginals, or relations changes the problem. Without cross-node relations or support restrictions, Minimum equals Independent; with a unique feasible law, it equals that law, including full coupling when required. For any alternative feasible coupler, the information-projection inequality gives

$$\mathrm{TC}(\pi) \geq \mathrm{TC}(\pi_{\min}) + D_{\mathrm{KL}}(\pi \,\|\, \pi_{\min})\,. \tag{6}$$

The difference $\mathrm{TC}(\pi) - \mathrm{TC}(\pi_{\min})$ is the *information surcharge*, whether a prompt, decoder, or dataset constructor produced π. For finite candidate sets, standard relative-interior conditions give a unique optimum and soft constraints trace a nondecreasing convex information frontier. For forests with edge-factorized non-node constraints, edge fitting, storage and sampling scale linearly when local alphabets and per-edge solve costs are bounded. Appendices A and C give the conditions, proofs, computation, and tests up to $10{,}000$ nodes.

4 The Joint Law Is Learner-Visible

Does a finite learner respond to this target freedom? We first change two-worker response frequencies at matched local content, check common update normalization, then test graded laws, a second relation and learner family.

4.1 Exact-Prompt Joint-Law Intervention

We instantiate the two-worker task from Section 2 with four legal implementation styles: a list comprehension, `filter` with a lambda, an explicit loop, and a generator expression. The specification constrains complementary orientation but not how styles co-vary across workers. Training covers 336 specifications with one fixed prompt each. Under every prompt, both arms contain the same 32 constructed legal response modes and the same worker-local strings, ASTs, and token IDs. Minimum repeats every style pair twice; *Surplus* repeats each same-style pair five times and each cross-style pair once, within each orientation. Same-style mass therefore changes from 0.25 to 0.625, adding 0.451 bits of task-free dependence without deleting support or changing a worker marginal. Split each response into one example per worker and the local training multisets are identical.

Three paired Qwen3-8B runs use response-only full-parameter SFT for one epoch of 672 updates. Model revision, prompts, epoch-total source and supervised tokens, loss, recipe, and checkpoint selection match. These original runs do not match the realized supervised-token denominator at every update; Section 4.3 reports a one-seed common-denominator rerun for the headline pair. The complete construction, balance audit, and optimizer replay are in Appendices C.2 and D.11.

4.2 The Injected Pairing Structure Transfers to Held-Out Specifications

We evaluate 32 held-out specifications built from unseen moduli 29–32: new instances of the same task family, so a contrast on them measures transfer of the injected pairing rule within the family rather than recall of training prompt–response pairs. The primary endpoint, *joint-law transfer*, draws 64 unconstrained continuations per specification, executes and classifies each program, and measures the Surplus-minus-Minimum difference in same-style rate with arms paired within specification.

Figure 2 shows the pairing pattern in every seed, with all generated programs valid. The same-style contrast is 47.1–47.8 points; subtracting each student's marginal-implied probability leaves 46.5–47.1, ruling out marginal concentration as its main source.

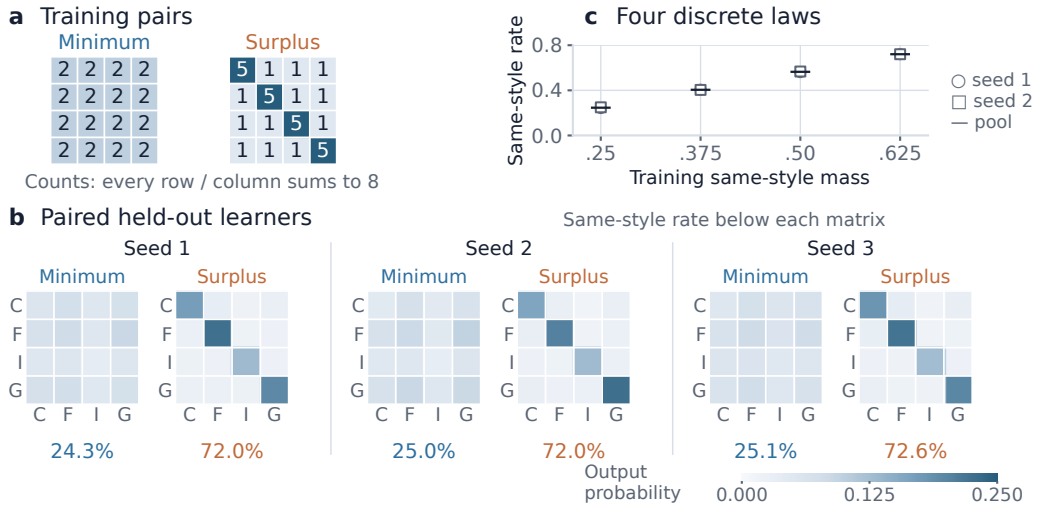


Figure 2: **A pairing pattern becomes learned output structure.** (a) Exact training counts per prompt and orientation. (b) Three paired seeds, 2,048 held-out generations per arm, one probability scale. Matrix rows/columns are worker-1/2 styles: comprehension (C), functional (F), imperative (I), generator (G). Seeds follow Table 6. (c) Two seeds at four discrete counterbalanced laws, reusing endpoint students.

Every seed–specification contrast is positive; conditioning on required orientation changes the gap by at most one point. Effective support—the entropy-equivalent number of emitted style pairs—contracts from about 14 to 8.5. Uniform-target NLL, permutation-calibrated MI, explicit controls, and complete matrices are in Appendices C.6, D.2, and D.3.

4.3 The Learner Effect Survives a Common Update Denominator

Equal epoch token totals do not equalize each update's label coefficient in the frozen training stack. At seed 20260827, rerunning both arms with the same fixed supervised-token denominator at every update gives same-style rates of 0.244 under Minimum and 0.721 under Surplus: a 47.75-point contrast with paired 95% interval $[45.36, 50.10]$ points. Both arms return $2{,}048/2{,}048$ valid programs under the otherwise unchanged protocol. Unequal update normalization is therefore not sufficient to explain this paired-seed effect. This is one rerun of an existing seed; the graded, second-relation and second-family comparisons remain data-level. Full protocol and own-marginal-adjusted results are in Appendix C.3.

4.4 The Same Variable Is Graded and Designable

Two counterbalanced interior laws vary style-pair counts between the endpoints while retaining the controls above (Appendix C.7). Held-out same-style generation increases at every adjacent level in both seed pairs (Figure 2c). Across the four discrete laws, a linear summary has slope 1.265 (95% interval $[1.208, 1.321]$); own-marginal adjustment preserves the ordering. Appendix C.7 gives the construction, exact values and conditional intervals.

4.5 A Second Relation and a Second Learner Family

Two further repetitions test the intervention beyond one relation and one learner family. We repeat it with a four-component executable interface relation and a fixed pool of 48 LLM-generated programs. Arms match complete legal support, component marginals, candidate-token multisets, rows, and epoch-total tokens; only legal-tuple multiplicities change. Across three paired Qwen3-8B seeds, the own-marginal-adjusted all-four-same contrasts are $+0.242$, $+0.240$, and $+0.249$, and every seed–wording contrast is positive. A single precommitted Llama-3.1-8B-Instruct pair likewise changes same-style behavior, providing second-family existence rather than an effect distribution.

Appendix C.8 gives the construction, intervals, and executable checks; Appendix C.9 gives the bounded Llama result.

5 The Joint Validation Law Reverses Checkpoint Selection

The second consequence needs no new training run. Holding the checkpoints, scored responses, tokenizer, and metric fixed, we change only the joint law under which complete-response likelihood is averaged, and the ranking reverses. Since likelihood-based selection on grouped data requires such a weighting, the question is how much of the law one fixed decision needs.

5.1 Same Rows, Opposite Rankings

Under Minimum weights, the Minimum checkpoint has lower complete-response NLL; under Surplus weights, the Surplus checkpoint does (Figure 3, left). Both scorings read the same 1,024 frozen rows—32 legal responses for each of 32 held-out specifications—and differ only in how those rows are weighted. Minimum weights every tuple by 2; Surplus weights same-style tuples by 5 and cross-style tuples by 1. Both place mass 64 on every specification, preserve the same worker marginals and legal support, and have the same mean weighted response length of 162.5 tokens. Under the local contract of Section 4.1, the scorings are audit-equivalent.

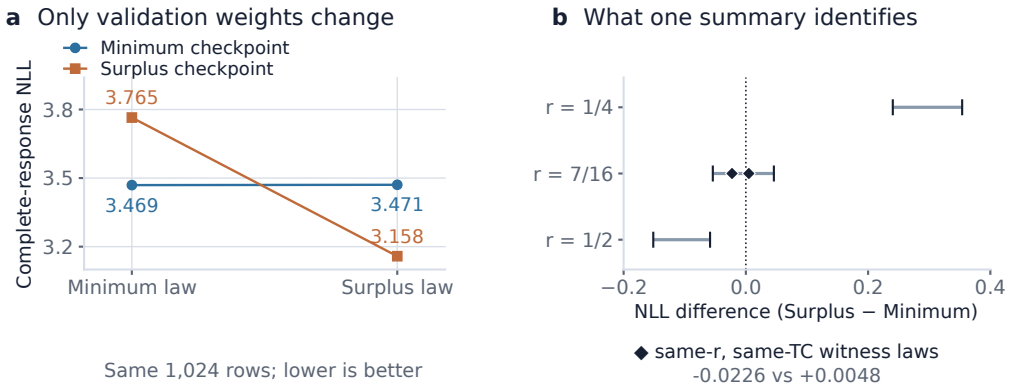


Figure 3: **The same scored rows permit opposite selections.** Left: validation weights reverse the matched-denominator checkpoints' complete-response NLL ranking. Right: finite-population compatible-law ranges at fixed same-style mass r; two post-hoc laws at $r = 7/16$ share even total correlation yet select opposite checkpoints. These ranges are not sampling intervals.

The $32/32$ reversal holds in the three original seed pairs and the matched-denominator rerun at seed 20260827: four pairs covering three seeds, not four independent replications. Equation (3) compares population targets; the experiment adds that finite checkpoints realize that direction on every specification. Each stated weighting determines a likelihood ranking, not external correctness; the local contract leaves open which weighting was intended. Appendix C.4 reports all four pairs.

5.2 How Much of the Law Does One Decision Need?

A fixed risk can be constant over many compatible laws, and a ranking is identified once the sharp range of its risk difference has one sign. For the matched-denominator pair, the recorded prompt, orientation, and style margins alone leave the ranking unidentified. Adding a same-style mass of 25% or 50% gives a sign-stable certificate, but that summary is not generally sufficient: two exact integer laws with mass $7/16$, the same local margins, full positive support, and the same 0.451-bit total correlation select opposite checkpoints (Figure 3, right). A certificate can thus suffice for one prespecified comparison, while the complete law supports arbitrary scores and future comparisons. These post-hoc finite-population bounds are specific to this pair, metric, tokenization, and rows. Appendix C.5 gives the sharp ranges and witness laws; the bounds use established methods (Maia Polo et al., 2024).

6 When the Joint Law Reaches a Decision

We next train on two complete programs in one response to the same competitive-programming task. Each training task uses one exact prompt and six accepted programs. *Cross* upweights pairings drawn from different program sources; the *treatment* upweights same-source pairings. Both retain all 30 legal ordered pairs, the same programs at the same per-slot frequencies, and matched rows, tokens, and 216 fixed-denominator updates at two paired seeds.

The design changes *which* programs partner, not the atom-level amount of dependence: every program has the same partner-count multiset $[0, 1, 1, 3, 3, 3]$ in both arms, and atom-level conditional entropy, joint entropy, and total correlation agree. Source labels never enter the model input, but broader token- and strategy-level similarity is not separately balanced: the intervention identifies the complete-response law as a bundle. We first score a prespecified archived-program vote, then separately generate a new second program behind an identical actual first prefix.

6.1 A Frozen Consumer Changes Decisions on Unseen Inputs

Using a consumer fixed before the new treatment training, we compare two newly trained treatment checkpoints with existing Cross checkpoints at the same seeds. The rule takes both programs from one response and a third from another, accepting an output when two of three agree exactly. On 80 tasks' public examples, the treatment-minus-Cross wrong-acceptance contrast has a conservative lower bound of $+4.04$ points (descriptive paired-task 95% interval $[+0.92, +7.30]$), with the same direction at both seeds (Appendix D.8).

The same archived programs were then scored on 1,186 official-domain inputs from 75 equally weighted source-qualified tasks, absent from the public examples. Wrong acceptance rises $+4.31$ points (descriptive paired-task 95% interval $[+1.46, +7.34]$); total acceptance rises $+4.98$ but correct acceptance only $+0.67$ (Appendix D.9).

Two ballots come from one response, so matching wrong outputs there already form a majority. Figure 4 partitions the contrast: the increase comes from these within-response votes, partly offset by votes split across responses. Three-of-four needs another matching vote; its contrast is only $+0.18$ points $[-1.03, +1.37]$, not established. This is the rule's arithmetic, not a training mediator (Appendix D.9).

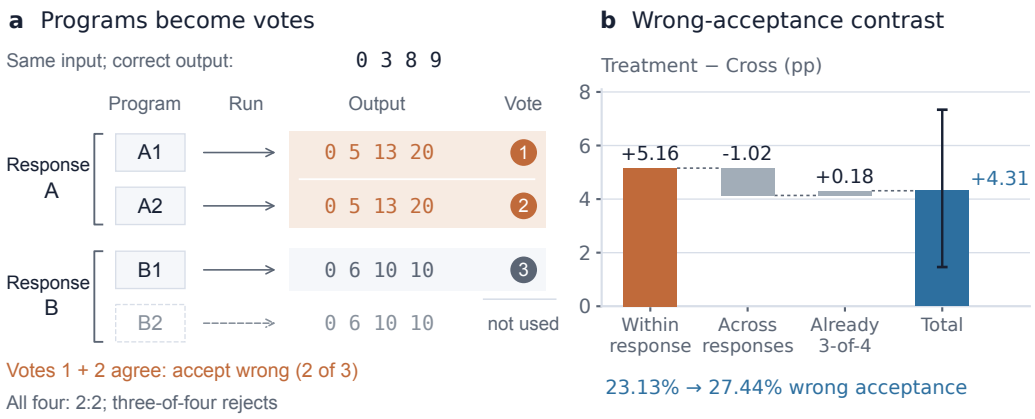


Figure 4: **One response can supply a majority of wrong votes.** (a) Archived treatment case: each complete output on the same input is one vote. The rule takes both A programs and a uniformly drawn B program (B1 shown). (b) Wrong-acceptance contrast over 75 equally weighted tasks by vote structure; the total whisker is a paired-task 95% interval conditional on two fixed training seeds.

6.2 A Second Attempt: More Agreement, Less Rescue

Separately, both matched checkpoints receive the *same actual first-program prefix* and generate only the second. A result-blind rule admits 2,324 of 2,400 prefixes on the same 75 tasks; one of 4,648 continuations remains unknown and is bounded throughout (Appendix D.10).

Figure 5 separates two units. *Per input*, we measure how often the second program matches its actual first program's incorrect output, minus a different-prefix baseline in the same cell. Treatment raises this same-wrong excess by 5.59–5.68 points (outer paired-task 95% interval $[4.00, 7.31]$), positive at both seeds; the baseline moves less than 0.22 point.

Per program, success means passing every unseen input of a task. The second program's overlap with a correct first grows, while its rescue of a failed first shrinks (Figure 5, right). With the first program fixed, that rescue loss is the entire 0.95–1.00-point decline in at-least-one success (outer interval $[-1.82, -0.21]$), about one quarter of Cross's incremental rescue share. Growing overlap offsets much of the rescue loss: the second program's own success rate changes by only -0.12 to -0.07 point, with an interval spanning zero. The comparison reveals reduced complementarity behind a fixed first attempt, without establishing marginal equivalence.

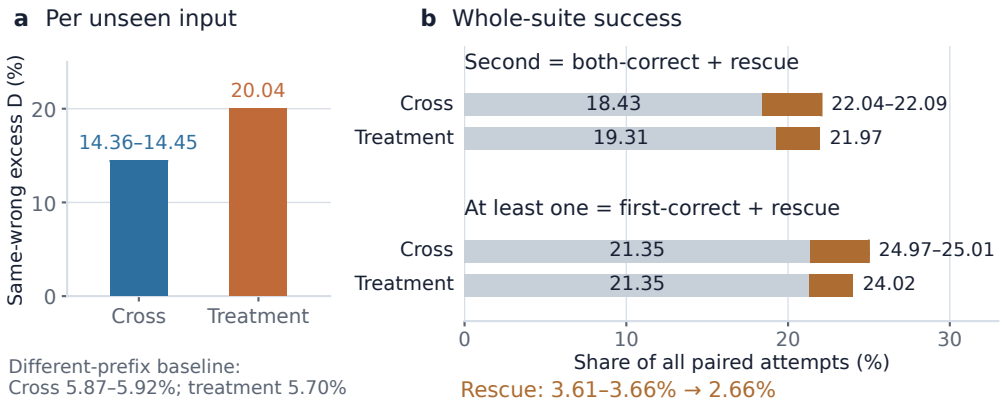


Figure 5: **A fixed first program reveals less incremental rescue.** Left: input-level same-wrong excess after subtracting a different-prefix baseline. Right: whole-suite success shares of *all* paired attempts, not conditional rates among first failures. The first programs are identical across arms, so lost rescue equals lost at-least-one success. Printed ranges bound one unknown continuation, not sampling uncertainty.

6.3 Behavioral Reach Is Not a Correctness Guarantee

A refinement variant asks the model to rewrite the two workers to match requested ascending or descending orders. In a preregistered Qwen3-8B pair, prompts, atoms, support, recipe and all 672 update denominators match; only response frequencies change. Minimum returns $103/512$ refined-valid outputs against the same-order-heavy Coupled arm's $0/512$ ($+20.12$ points $[17.77, 22.46]$), yet the compositional interaction is -23.83 points $[-29.69, -17.97]$, opposite the required direction (Appendix D.1).

Two released-program carriers likewise separate composition from correctness, at matched inventories and fixed update denominators. In CodeContests, source-dissimilar pairing reduces source/AST repeats, while two-attempt any-success changes -1.28 points $[-4.27, +1.71]$. In a multi-source AtCoder carrier, Cross-heavy pairing cuts same-style responses from 81.6% to 32.0%, while any-correct changes -3.13 points $[-9.87, +3.62]$. Neither establishes its registered correctness gain (Appendix Table 12; protocols in Appendices D.4 and D.5).

7 Related Work

Dependent generation and grouped SFT. Coupled decoding, coordinated sampling, packing, and grouped SFT can change candidate coverage, atoms, objectives, or co-location (De Mijolla et al., 2025; Ravfogel et al., 2025; Zhao et al., 2026; Handa et al., 2026; Li et al., 2026; Krell et al., 2021; Wang et al., 2025; Dong et al., 2025; Li et al., 2025; Fan et al., 2026; Schaffelder & Gatt, 2026). We instead change complete same-task response frequencies while leaving local supervision unchanged.

Separate losses, complete records, and correlated failure. The population objective of separate-loss multi-response training is pairing-invariant at fixed answer marginals (Amin et al., 2026); that of one autoregressive response is not. Grouped records and data cards exist (Min et al., 2020; Pushkarna et al., 2022); solution-diversity and correlated-failure studies are adjacent (Lee et al., 2026; Fan et al., 2026; Tieman & Markou, 2026). SCoRe studies retaining correct first answers versus rescuing incorrect ones in sequential correction (Kumar et al., 2024). We isolate the effect of fixed-atom training-row pairing and read out new second attempts behind an identical first-program prefix.

Label dependence in classical prediction. Marginal-versus-joint risk, minimum-information laws, entropy projection, transport, and dependence-aware bounds are classical (Dembczyński et al., 2012; Read et al., 2011; Meeuwissen & Bedford, 1997; Bedford & Wilson, 2014; Csiszár, 1975; Deming & Stephan, 1940; Sinkhorn, 1964; Haasler et al., 2021b;a; Albergo et al., 2024; Chatterjee et al., 2025). We apply established compatible-law selection bounds (Maia Polo et al., 2024) to a constructor-controlled target law.

8 Discussion and Limitations

What the evidence establishes. The controlled comparisons expose a reproducibility gap: local audits can agree while grouped training targets and likelihood-based choices differ. The released-program tests give that choice a bounded decision consequence, through a specified vote and the rescue supplied by a second attempt.

Where the evidence stops. The clean learner effect has a common update denominator at only one existing seed, and selection fixes one scored population. Released-program intervals condition on four checkpoints, two seeds, one task family, sandbox and voter; the three-of-four comparison is unconfirmed. Fixed-prefix continuations measure conditional association, not a copying mechanism or mediation of the archived vote. Bounded missingness and sensitivity to later timeouts protect the fixed-prefix result's direction, not an environment-independent magnitude (Appendix D.10).

The utility tests in Section 6.3 show why behavioral reach alone cannot select a useful law. Its value depends on what the application needs from the group, not simply on how much dependence remains.

The grouped-supervision contract. Immutable complete rows retain empirical frequencies. When a release instead specifies local outputs and regenerates or reassembles groups, reproducibility requires a replayable complete-response law. Likelihood-based selection also needs validation weights; for a prespecified comparison, a sign-stable certificate can sometimes suffice.

9 Conclusion

In grouped autoregressive supervision, the tuple—not merely its component answers—is the target. Its frequencies change learned behavior, likelihood selection, a specified program vote, and second-attempt rescue despite matched local supervision. Minimum-sufficient coupling makes this construction choice auditable without prescribing a universally useful law. Preserve complete grouped rows or a replayable response law, including validation weights.

AI Use Statement

Generative AI tools supported literature review, refinement of research ideas, experimental design and implementation, generation of synthetic research materials, mathematical derivations, analysis

and interpretation of results, and manuscript preparation and polishing. The authors led the research, directed the use of these tools, and made the final scientific decisions. The authors take responsibility for the final content.

Reproducibility statement. The appendix provides proofs, experimental protocols, full result tables, construction and execution audits, and reproducibility details.

REFERENCES

Michael S. Albergo, Nicholas M. Boffi, Michael Lindsey, and Eric Vanden-Eijnden. Multimarginal generative modeling with stochastic interpolants. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f72af04ff6852f2669b2ed620c3c86c8-Abstract-Conference.html.

Hasan Amin, Kian Ahrabian, Ming Yin, and Rajiv Khanna. Escaping the mode lottery: Multi-response training improves language model generalization, 2026. URL https://arxiv.org/abs/2606.00544.

Tim Bedford and Kevin J. Wilson. On the construction of minimum information bivariate copula families. *Annals of the Institute of Statistical Mathematics*, 66(4):703–723, 2014. doi: 10.1007/s10463-013-0422-0.

Sagnik Chatterjee, Manuj Mukherjee, and Alhad Sethi. Generalization bounds for dependent data using online-to-batch conversion. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2152–2160, 2025. URL https://proceedings.mlr.press/v258/chatterjee25b.html.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158, 1975. doi: 10.1214/aop/1176996454.

Damien De Mijolla, Hannan Saddiq, and Kim Moore. Waste not, want not: Recycled Gumbel noise improves consistency in natural language generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5662–5686, 2025. doi: 10.18653/v1/2025.naacl-long.292. URL https://aclanthology.org/2025.naacl-long.292/.

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hüllermeier. On label dependence and loss minimization in multi-label classification. *Machine Learning*, 88(1–2):5–45, 2012. doi: 10.1007/s10994-012-5285-8.

W. Edwards Deming and Frederick F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444, 1940. doi: 10.1214/aoms/1177731829.

Jiancheng Dong, Lei Jiang, Wei Jin, and Lu Cheng. Threshold filtering packing for supervised fine-tuning: Training related samples within packs. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4422–4435, 2025. doi: 10.18653/v1/2025.naacl-long.226. URL https://aclanthology.org/2025.naacl-long.226/.

Zhiyuan Fan, Guanqiao Chen, Yanyi Huang, Mingkuan Zhao, Dadi Guo, and Yi R. Fung. Learning diverse responses with prefix-conditioned supervised fine-tuning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 247–276, 2026. doi: 10.18653/v1/2026.acl-long.9. URL https://aclanthology.org/2026.acl-long.9/.

Isabel Haasler, Axel Ringh, Yongxin Chen, and Johan Karlsson. Multimarginal optimal transport with a tree-structured cost and the Schrödinger bridge problem. *SIAM Journal on Control and Optimization*, 59(4):2428–2453, 2021a. doi: 10.1137/20M1320195.

Isabel Haasler, Rahul Singh, Qinsheng Zhang, Johan Karlsson, and Yongxin Chen. Multi-marginal optimal transport and probabilistic graphical models. *IEEE Transactions on Information Theory*, 67(7):4647–4668, 2021b. doi: 10.1109/TIT.2021.3077465.

Divij Handa, Mihir Parmar, Aswin RRV, Md Nayem Uddin, Hamid Palangi, and Chitta Baral. GuidedSampling: Steering LLMs towards diverse candidate solutions at inference-time. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ea9cfce070cebac95fec35f35b298e30-Abstract-Conference.html.

Mario Michael Krell, Matej Kosec, Sergio P. Perez, and Andrew Fitzgibbon. Efficient sequence packing without cross-contamination: Accelerating large language models without impacting performance, 2021. URL https://arxiv.org/abs/2107.02027.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, et al. Training language models to self-correct via reinforcement learning, 2024. URL https://arxiv.org/abs/2409.12917.

Sangmook Lee, Minbeom Kim, Jeonghye Kim, Dohyung Kim, Sojeong Rhee, and Kyomin Jung. Are we measuring strategy or phrasing? the gap between surface- and approach-level diversity in LLM math reasoning, 2026. URL https://arxiv.org/abs/2606.29985.

Ming Li, Pei Chen, Chenguang Wang, Hongyu Zhao, Yijun Liang, YuPeng Hou, Fuxiao Liu, and Tianyi Zhou. Mosaic-IT: Cost-free compositional data synthesis for instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25287–25318, 2025. doi: 10.18653/v1/2025.findings-acl.1297. URL https://aclanthology.org/2025.findings-acl.1297/.

Yilong Li, Suman Banerjee, and Tong Che. Cast a wider net: Coordinated Pass@K policy optimization for code reasoning, 2026. URL https://arxiv.org/abs/2605.27000.

Felipe Maia Polo, Subha Maity, Mikhail Yurochkin, Moulinath Banerjee, and Yuekai Sun. Weak supervision performance evaluation via partial identification. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-4309. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f4c6bec746b0aeca8c2cd15096f1ad1f-Abstract-Conference.html.

A. M. H. Meeuwissen and Tim Bedford. Minimally informative distributions with given rank correlation for use in uncertainty analysis. *Journal of Statistical Computation and Simulation*, 57 (1–4):143–174, 1997. doi: 10.1080/00949659708811806.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 5783–5797, 2020. doi: 10.18653/v1/2020.emnlp-main.466. URL https://aclanthology.org/2020.emnlp-main.466/.

Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. Data cards: Purposeful and transparent dataset documentation for responsible AI, 2022. URL https://arxiv.org/abs/2204.01075.

Shauli Ravfogel, Anej Svete, Vésteinn Snæbjarnarson, and Ryan Cotterell. Gumbel counterfactual generation from language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e15790966a4a9d85d688635c88ee6d8a-Abstract-Conference.html.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011. doi: 10.1007/s10994-011-5256-5.

Max Schaffelder and Albert Gatt. Synthetic eggs in many baskets: The impact of synthetic data diversity on LLM fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7265–7293, 2026. doi: 10.18653/v1/2026.findings-acl.360. URL https://aclanthology.org/2026.findings-acl.360/.

Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35(2):876–879, 1964. doi: 10.1214/aoms/1177703591.

Ross Tieman and Evan Markou. Inferred generative-process diversity predicts correlated failure across language models, 2026. URL https://arxiv.org/abs/2609.03422.

Shuhe Wang, Guoyin Wang, Yizhong Wang, Jiwei Li, Eduard Hovy, and Chen Guo. Packing analysis: Packing is more appropriate for large models or datasets in supervised fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4953–4967, 2025. doi: 10.18653/v1/2025.findings-acl.256. URL https://aclanthology.org/2025.findings-acl.256/.

Ruiqing Zhao, Fengzhi Li, Yuan Zuo, Rui Liu, Yansong Liu, Yunfei Ma, Fanyu Meng, and Junlan Feng. Strategy-aware optimization modeling with reasoning LLMs. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.02545.

A Proofs and Derivations

A.1 Unique Minimum-Information Coupling

Let $Y = (Y_1, \ldots, Y_m)$ range over a finite product space, let the frozen node marginals be p_i, and define the independent reference $q(y) = \prod_i p_i(y_i)$. Let Ω be the declared legal support. The feasible set is

$$\mathcal{F}_\tau = \{\pi : \operatorname{supp}(\pi) \subseteq \Omega, \quad \pi_i = p_i \ \forall i, \quad \mathbb{E}_\pi[r_e(Y_e)] \geq \tau_e \ \forall e\}.$$

For every $\pi \in \mathcal{F}_\tau$,

$$D_{\mathrm{KL}}(\pi \| q) = \sum_i H(p_i) - H(\pi) = \mathrm{TC}(\pi),$$

so KL projection onto $\mathcal{F}_\tau$ is exactly minimum total correlation at the requested marginals and relation levels.

If $\mathcal{F}_\tau$ is nonempty, it is a compact convex subset of the finite probability simplex. KL is lower semicontinuous and strictly convex on the feasible support where q is positive, which gives existence and uniqueness of $\pi_{\min}$. Write $\Omega_{\mathcal{F}} = \bigcup_{\pi \in \mathcal{F}_\tau} \operatorname{supp}(\pi)$. If the marginal equalities admit a feasible point in their relative interior on $\Omega_{\mathcal{F}}$ and every non-forced soft moment inequality is strictly feasible there, the Lagrangian for normalization, marginal equalities, and relation inequalities yields

$$\pi_{\min}(y) = \mathbf{1}\{y \in \Omega_{\mathcal{F}}\} q(y) 2^{\sum_e \lambda_e r_e(y) + \sum_i \alpha_i(y_i) - A}, \qquad \lambda_e \geq 0.$$

Here the potentials and multipliers are normalized to base-two log units, matching the information quantities above. The node potentials α_i restore the frozen marginals, and complementary slackness sets $\lambda_e = 0$ for every inactive relation. For finite supports, Sinkhorn/IPF computes the node potentials for fixed relation multipliers and a dual update adjusts the multipliers. Hard legality is represented by restricting to Ω; the effective support $\Omega_{\mathcal{F}}$ additionally removes tuples that the frozen marginals and relations force to zero, rather than representing either condition by taking a finite multiplier to a limit. Infeasibility is reported rather than repaired by moving a marginal or threshold.

A.2 Information Surcharge and Budget Curve

Because $\pi_{\min}$ is the I-projection of q onto a closed convex set, the Bregman Pythagorean inequality gives, for every feasible π,

$$D_{\mathrm{KL}}(\pi \| q) \geq D_{\mathrm{KL}}(\pi \| \pi_{\min}) + D_{\mathrm{KL}}(\pi_{\min} \| q).$$

Using the fixed-marginal identity above,

$$\mathrm{TC}(\pi) \geq \mathrm{TC}(\pi_{\min}) + D_{\mathrm{KL}}(\pi \| \pi_{\min}).$$

For an affine equality family the relation is an equality; with inequality constraints, direct substitution of the KKT form gives the exact gap

$$\mathrm{TC}(\pi) - \mathrm{TC}(\pi_{\min}) - D_{\mathrm{KL}}(\pi \| \pi_{\min}) = \sum_e \lambda_e^* \left\{\mathbb{E}_\pi[r_e] - \mathbb{E}_{\pi_{\min}}[r_e]\right\}.$$

Thus equality holds exactly when every constraint with $\lambda_e^* > 0$ has the same moment under π and $\pi_{\min}$. Dependence beyond Minimum is lower-bounded by a nonnegative divergence from the specification-relative reference; the statement does not depend on how the candidate law was generated.

For one soft relation target, define

$$C(\tau) = \min_{\substack{\pi : \operatorname{supp}(\pi) \subseteq \Omega, \ \pi_i = p_i, \\ \mathbb{E}_\pi[r] \geq \tau}} \mathrm{TC}(\pi).$$

If $\tau_2 \geq \tau_1$, the feasible set at τ_2 is nested inside the one at τ_1, so C is nondecreasing. Convexity follows from the convex value function of a linear right-hand-side constraint. At regular differentiable points, the envelope theorem gives $C'(\tau) = \lambda^*(\tau) \geq 0$, where λ^* is the optimal relation multiplier. This is the information-budget interpretation used in the main text.

A.3 Shared and Local Degrees of Freedom

Suppose node i emits (S_i, L_i) with $S_i \sim p_S$, $L_i \sim p_L$, and $S_i \perp L_i$. The task requires $S_1 = \cdots = S_m$ and imposes no relation on the L_i. Minimum samples one shared S and samples $L_1, \ldots, L_m$ independently, giving

$$\mathrm{TC}(\pi_{\min}) = (m-1)H(S).$$

A global law that shares both coordinates instead has

$$\mathrm{TC}(\pi_{\mathrm{global}}) = (m-1)\{H(S) + H(L)\},$$

and therefore pays the exact surcharge $(m-1)H(L)$. With N sampled groups, estimating a Bernoulli local rate has

$$\mathrm{Var}(\hat{\theta}_{\mathrm{Global}}) = \frac{\theta(1-\theta)}{N}, \qquad \mathrm{Var}(\hat{\theta}_{\mathrm{Minimum}}) = \frac{\theta(1-\theta)}{Nm}. \tag{7}$$

The resulting variance ratio is m despite equal node marginals and perfect required-relation fidelity. The same construction gives the two zero boundaries: with no cross-node relation, and no additional support restriction, the product law is feasible and Minimum equals Independent; if the specification leaves a unique feasible joint law, Minimum equals that law, including the fully coupled case when required by the relation.

A.4 Exact Factorization on Forests

Let the typed-relation graph be a forest, with every non-node legality and moment constraint factorized over its edges. For each edge $e = (u, v)$, let μ_e^* be the maximum-entropy feasible pair law with endpoint marginals p_u, p_v. Then

$$\pi_F^*(y) = \frac{\prod_{e=(u,v)} \mu_e^*(y_u, y_v)}{\prod_{i:\, d_i>0} p_i(y_i)^{d_i-1}} \prod_{i:\, d_i=0} p_i(y_i),$$

so each isolated node contributes exactly one $p_i(y_i)$ factor. The tree entropy inequality states

$$H(Y) \le \sum_{e=(u,v)} H(Y_u, Y_v) - \sum_i (d_i - 1)H(Y_i),$$

with equality for the tree-Markov product above. Since node entropies are fixed, maximizing each feasible edge entropy independently maximizes the global entropy. Consequently

$$\mathrm{TC}(\pi_F^*) = \sum_{e=(u,v)} I_{\mu_e^*}(Y_u; Y_v),$$

and π_F^* is the unique forest-structured Minimum. Pairwise IPF computes the edge laws; root sampling followed by ancestral sampling gives linear storage and sampling cost for fixed local support. The forest creates a linear number of independent edge-fitting subproblems; total fitting time is linear in the number of edges only when the local alphabet and per-edge solve cost are uniformly bounded.

B Candidate Construction and Executable Oracles

B.1 Real LLM Candidates in the Transparent Tasks

The four-component pipeline contains successive Python components for increment, multiplication, even filtering, and checksum. Each of the three interfaces uses either a tuple or a dictionary carrying a list; adjacent components must agree on the interface, while control flow is unconstrained. We queried `qwen3.8-max` once for each of 12 role–interface conditions and requested four candidates, producing 48 programs. Calls used temperature 0.8, thinking enabled with a 2,048-token budget, and a 3,072-token completion limit. The parser accepted exact JSON or JSON with one surrounding Markdown fence removed. No candidate was selected, repaired, replaced, or regenerated: every program had to compile, satisfy its component contract, and pass the end-to-end pipeline oracle. AST precedence classified the resulting candidates as imperative, functional, comprehension, or builtin.

The complementary-coverage task was generated independently. Each of two workers receives the even or odd shard, and a valid group assigns opposite shards so that outputs are disjoint and

their multiset union exactly recovers the input. For each worker–shard condition, `qwen3.8-max` produced comprehension, functional, imperative, and generator implementations at temperature 0.7 with thinking enabled. All 16 programs passed tests for empty inputs, negative numbers, duplicates, ordering, and mixed parity; none was repaired, selected, or replaced. These two tasks share neither relation, prompt, candidate pool, nor downstream transparent learner.

B.2 Exact-Prompt Neural Construction

The neural intervention reuses the complementary-residue oracle but constructs canonical responses rather than selecting among model-generated candidates. Training covers moduli 8–28, four residue families, and four offsets, giving 336 semantic specifications. One prompt realization is frozen per specification before response assignment. For each exact prompt, the response support is the Cartesian product of two legal orientations and 16 worker style pairs: list comprehension, `filter` with a lambda, explicit loop, or generator expression at each worker. Every response is parsed, executed, checked for disjoint outputs and exact multiset coverage, and classified by AST. The held-out set uses unseen moduli 29–32, four residue families, and two offsets, giving 32 semantic specifications and $1{,}024$ uniformly weighted canonical responses.

Minimum repeats every style pair twice within each orientation. Surplus repeats each of the four diagonal pairs five times and each of the twelve off-diagonal pairs once. Both arms therefore contain every one of the 32 legal modes under every exact prompt, while each worker uses each style eight times per orientation. Only pair multiplicity changes. The graded intervention of Section 4.4 changes this count matrix alone: its two interior laws place three and four copies in each diagonal cell, with counterbalanced off-diagonal fills. Appendix C.7 gives the construction and balance checks.

C Balance, Evaluation, and Solver Audits

C.1 Transparent-Consumer Protocols

Each row of Table 1 fits its consumer to one finite sample drawn from the joint law under test and scores it exactly against a fixed test law; repetitions are paired across laws within a row.

Analytic construction. A training set consists of 64 groups of eight nodes, hence 512 observations of the task-free bit. The consumer is the empirical rate over these labels, clipped only when the log loss is evaluated, and is scored under the exact Bernoulli test law of the free bit. Reported means and variances use $4{,}000$ repetitions with seed 20260820.

Interface pipeline. A training set consists of 32 four-component groups (128 component selections). The consumer is an unregularized logistic model with an intercept and a single slope; its feature is the exact log-odds of each candidate pool's non-comprehension rate, and its label indicates whether a selected candidate is non-comprehension. Test NLL is computed exactly over the twelve candidate pools. Reported means and variances use $5{,}000$ paired repetitions with seed 20260820.

Complementary coverage. A training set consists of 32 two-worker groups, hence 64 strategy labels. The consumer estimates the four-strategy distribution by Jeffreys-smoothed empirical frequencies $(n_k + \frac{1}{2})/(64 + 2)$ and is scored by exact NLL under the uniform four-strategy population. Reported means and variances use $5{,}000$ paired repetitions with seed 20260821.

Table 1: Node-separable consumers at matched node marginals. Fidelity is shown for Independent and the feasible laws; remaining columns compare Global with Minimum at fidelity one.

Setting	Fidelity (Indep. / feasible)	TC (G−M)	Δ NLL (G−M)	Variance (G/M)
Analytic, $m = 8$	$.017/1.000$	6.169 bits	$+.00730$	$8.23\times$
LLM interface	$.125/1.000$	3.9528 bits	$+.01041$	$5.99\times$
LLM coverage	$.500/1.000$	2.000 bits	$+.02682$	$5.09\times$

In the analytic construction, Independent preserves the node marginals but attains relation fidelity only 0.01745. Minimum and Global attain fidelity one, while Global adds $7H(L) = 6.169$ bits and

inflates local-rate estimator variance by $8.23\times$, close to the factor $m = 8$ predicted by Eq. (7). The population target is unchanged. The same pattern appears for the two executable relations: all 48 interface and 16 coverage candidates pass their frozen oracles, while Global raises mean test NLL and sampling variance. For these small symmetric hard constraints, rejection sampling conditioned on feasibility realizes Minimum at average costs of eight and two proposals.

C.2 Exact-Prompt Balance Audit

Table 2: Data-level balance of the exact-prompt intervention. The assigned difference is legal style-pair multiplicity and task-free total correlation; the matched quantities do not imply equal optimizer-step normalization.

Quantity	Minimum	Surplus
Prompt bytes	identical; one frozen realization/specification	
Constructed legal support	all 32 modes	
Distinct prompt–responses	10,752	
Responses per prompt	64	64
Orientation counts	32/32	32/32
Per-worker style counts/orientation	8 each	8 each
Local-label multisets	identical strings, ASTs, and token IDs	
Program validity	100%	100%
Pair multiplicity/orientation	2 each	5 diagonal / 1 off-diagonal
Task-free style TC	0 bits	0.451 bits

Table 3 reports epoch-total tokenizer-level balance of the two arms. Counts are identical across the two arms even though individual canonical responses have different lengths; equality is verified on the realized training files rather than inferred from an equal-length assumption.

Table 3: Exact two-worker training-data audit, reported per arm. Both endpoint arms pass all four equality checks and contain zero truncated rows; the interior laws realize the same row, token, and support totals (Appendix C.7).

Quantity	Value per arm
Training rows	21,504
Source tokens	4,841,792
Supervised response tokens	2,418,176
Total tokens	7,259,968
Truncated rows	0

The audit additionally verifies exact per-prompt total-, source-, and supervised-token equality, identical model and tokenizer revisions, and a maximum sequence length of 1,024. Every reported run completes 672 updates, and evaluation reads checkpoint-672 only.

These totals do not establish optimizer-step balance. In the frozen stack, Qwen response-only loss sums supervised-token cross-entropy and divides by the global number of non-masked labels in each 32-row accumulation window. Deterministic replay of the source sampler over canonical JSONL order finds equal arm denominators in only 7, 6, and 4 of 672 updates for seeds 20260827–20260829. Across the 5,376 matched local atoms, mean relative differences in accumulated coefficients are 1.44–1.56% and the maximum is 8.93%. This does not show that optimizer weighting caused the behavioral contrast, but it rules out the stronger claim that the comparison already equalized optimizer-effective local-label weights. The original provider retained no sampler-index trace, so this audit is an exact source-stack replay conditional on canonical file order, not recovery of the historical minibatches.

C.3 Matched-Denominator Rerun

The rerun preserves the original two-worker data, paired seed 20260827, sampler, model revision, optimizer, 672-update schedule, checkpoint selection, and complete frozen evaluation. Its only

intended change is that both arms use the same fixed supervised-token denominator at every update. Each arm returns $2{,}048$ valid programs out of $2{,}048$ generations. Held-out same-style rates are 0.721191 under Surplus and 0.243652 under Minimum; the paired Surplus-minus-Minimum contrast is 0.477539, with a 95% specification-bootstrap interval of $[0.453613, 0.500977]$. The contrast point estimate is identical to the original run at this seed. The arm difference in marginal-implied same-style probability is 0.009331; the difference in excess same-style association is 0.468208, with interval $[0.444138, 0.491974]$.

C.4 Validation-Law Checkpoint Selection

The selection audit reuses immutable complete-response NLL rows from the three original seed pairs and the matched-denominator rerun at seed 20260827; the four pairs therefore cover three training seeds, not four independent replications. For every pair, the two checkpoint readbacks contain the same $1{,}024$ rows—32 legal responses for each of 32 specifications, with matching record keys, prompts, responses, lineage, response-token counts, and source keys—and no row is removed. Minimum assigns integer weight 2 to every tuple; Surplus assigns weight 5 to same-style and 1 to cross-style tuples. Both laws place mass 64 on every specification and share worker marginals, legal support, and mean weighted response length (162.5 tokens). Complete-response NLL is averaged under these weights.

Table 4: Law-weighted response-token NLL sums, in natural-log units per complete response. The difference is Surplus checkpoint minus Minimum checkpoint; lower is better.

Checkpoint pair	Validation law	Minimum ckpt.	Surplus ckpt.	Difference	Specs
Fixed denom., 20260827	Minimum	3.469184	3.764748	+0.295564	32/32
Fixed denom., 20260827	Surplus	3.470639	3.158157	−0.312482	32/32
Original 20260827	Minimum	3.469046	3.756118	+0.287073	32/32
Original 20260827	Surplus	3.470386	3.155670	−0.314716	32/32
Original 20260828	Minimum	3.470024	3.763636	+0.293613	32/32
Original 20260828	Surplus	3.473071	3.158309	−0.314762	32/32
Original 20260829	Minimum	3.470567	3.773595	+0.303028	32/32
Original 20260829	Surplus	3.467018	3.156205	−0.310813	32/32

For the matched-denominator pair, descriptive 95% specification-bootstrap intervals for the Surplus-minus-Minimum difference are $[0.294409, 0.296686]$ under the Minimum law and $[-0.314190, -0.310796]$ under the Surplus law. Every specification favors the checkpoint trained under the matching law in all four pairs. These are law-weighted complete-response NLLs and are not on the same scale as the uniform-target per-token NLL contrasts reported elsewhere. The result is a likelihood-based selection statement, not a correctness ranking or evidence that either law is generally preferable.

C.5 Decision-Specific Partial Identification

This post-hoc audit fixes the matched-denominator checkpoint pair, its tokenization and complete-response NLL metric, and the same $1{,}024$ scored legal responses. Let $\mathcal{V}$ be the family of nonnegative laws over the legal response set that preserve the recorded prompt weights, orientation weights within each prompt, and both workers' style marginals within each prompt and orientation. The primary family does not impose a positive per-tuple floor. For checkpoint h, define

$$R_h(\nu) = \sum_z \nu(z)\ell_h(z), \qquad \nu \in \mathcal{V}, \tag{8}$$

where $\ell_h(z)$ is its frozen complete-response NLL on row z. The compatible risk interval is

$$\mathcal{I}_h(\mathcal{V}) = \left[\inf_{\nu\in\mathcal{V}} R_h(\nu), \sup_{\nu\in\mathcal{V}} R_h(\nu)\right]. \tag{9}$$

For the Surplus and Minimum checkpoints, let $\Delta(\nu) = R_{\mathrm{S}}(\nu) - R_{\mathrm{M}}(\nu)$ and define $\mathcal{I}_\Delta$ analogously. A fixed risk is identified when $\mathcal{I}_h$ is a singleton. The pairwise ranking is identified when $\mathcal{I}_\Delta$ is strictly

positive or strictly negative, even if both individual risks remain non-unique. The complete target law is identified only when $\mathcal{V}$ is itself a singleton.

Under the local constraints alone, $\mathcal{I}_\Delta = [-0.920528, +0.745403]$. Fixing the overall same-style mass r gives

r	Sharp range of Δ	Selection
0.25	$[+0.240410, +0.353880]$	Minimum
0.4375	$[-0.054053, +0.045841]$	unidentified
0.50	$[-0.151470, -0.058577]$	Surplus

Positive intervals select Minimum and negative intervals select Surplus. Over the continuously constrained family, the full same-style ambiguity region is $r \in [0.402892, 0.465047]$, with equality allowing a tie. A separate sensitivity analysis that additionally requires each prompt-conditional tuple to have mass at least $1/64$ gives $[0.421242, 0.444940]$. This floor is not part of the primary family, and its range is not the ambiguity region above.

Two exact integer witness laws place same-style mass $r = 7/16$ in every prompt, reproduce the listed local margins, assign positive mass to every legal response, use 64 rows per prompt and $332{,}800$ total response tokens, and have task-free total correlation 0.451205 bits. Their score differences are respectively -0.022636 and $+0.004831$, so they select opposite checkpoints. The witnesses differ in their tuple weights and were optimized post hoc on the archived rows. They are existence witnesses, not recommended validation laws.

These finite-population bounds concern likelihood selection for this fixed pair. A different pair, metric, tokenization, or scored population requires a new certificate; the three original checkpoint pairs are not covered. We apply established partial-identification methods (Maia Polo et al., 2024) to the grouped-supervision contract.

C.6 Deterministic Local-Supervision and Structural Audits

The exact local audit groups labels by prompt, orientation, and worker. All $1{,}344$ paired groups contain 32 occurrences and four distinct local implementations per arm. The complete groupwise multisets match across arms under all three representations: literal program strings, parsed ASTs, and frozen-tokenizer token-ID sequences. The separate optimizer audit is in Appendix C.2.

Separately, a full-response serialization audit verifies the empirical bridge used in Eq. (3): for each of the 336 exact prompts, all 32 legal response strings remain distinct after frozen Qwen tokenization, yielding $10{,}752$ distinct prompt-conditioned response-token sequences and zero collision buckets in either arm. The two arms have exactly the same serialized support.

Table 5: Absolute levels and secondary contrasts behind Figure 2. Paired levels are Minimum / Surplus. Declared-target NLL is Surplus minus Minimum; explicit-control DID is off-diagonal minus diagonal adherence, Minimum minus Surplus. Slightly negative above-null values are null-centered point estimates compatible with zero.

Quantity	20260827	20260828	20260829
Same-style	.2427/.7202	.2495/.7202	.2510/.7256
Style MI above null (bits)	−.0057/.7196	−.0093/.7058	−.0050/.7307
Own-marg. excess	−.0116/.4597	−.0103/.4582	−.0022/.4632
Diag. adherence	.2109/.4844	.2500/.5000	.0391/.5000
Off-diag. adherence	.0156/.0000	.0000/.0000	.0312/.0000
Declared-target NLL contrast	+.001768 [.001748, .001789]	+.001808 [.001791, .001825]	+.001866 [.001846, .001886]
Explicit-control DID	+.2891 [.2474, .3333]	+.2500 [.2500, .2500]	+.4922 [.4531, .5208]

For seed 20260828, every held-out specification yields the same explicit-control contrast of 0.25 under greedy decoding, so every bootstrap resample agrees and the interval has zero width.

Across the full $21{,}504$ training rows, Minimum assigns count $1{,}344$ to every cell of the 4×4 style-pair matrix. Surplus assigns $3{,}360$ to each diagonal cell and 672 to each off-diagonal cell. The resulting same-style probabilities are 0.25 and 0.625. The students' free-generation matrices preserve

this structural difference across all three seeds (Table 6); rows and columns follow comprehension, functional, imperative, and generator order:

Table 6: Two-worker student style-pair counts from 2,048 valid generations per seed and arm. Each entry is a 4×4 matrix; Minimum / Surplus are shown.

Seed	Minimum	Surplus
20260827	$\begin{bmatrix}113 & 150 & 107 & 146\\ 141 & 164 & 106 & 181\\ 106 & 109 & 73 & 108\\ 142 & 156 & 99 & 147\end{bmatrix}$	$\begin{bmatrix}349 & 67 & 37 & 54\\ 53 & 460 & 40 & 43\\ 46 & 42 & 267 & 43\\ 50 & 53 & 45 & 399\end{bmatrix}$
20260828	$\begin{bmatrix}86 & 142 & 89 & 135\\ 136 & 177 & 95 & 199\\ 100 & 107 & 69 & 112\\ 138 & 180 & 104 & 179\end{bmatrix}$	$\begin{bmatrix}329 & 57 & 28 & 57\\ 62 & 419 & 35 & 49\\ 44 & 41 & 264 & 48\\ 60 & 43 & 49 & 463\end{bmatrix}$
20260829	$\begin{bmatrix}104 & 147 & 112 & 131\\ 132 & 168 & 118 & 164\\ 113 & 118 & 103 & 109\\ 127 & 158 & 105 & 139\end{bmatrix}$	$\begin{bmatrix}369 & 73 & 31 & 62\\ 64 & 453 & 32 & 41\\ 43 & 38 & 260 & 42\\ 56 & 41 & 39 & 404\end{bmatrix}$

Table 7 reports two further checks. Each seed is positive in every held-out modulus block. Separately, the NLL decomposition has the frequency-tracking signature expected from the injected law: Surplus improves diagonal responses and worsens the three-times-more-common off-diagonal target responses. Weighting the two components by their response-token counts recovers the positive aggregate NLL contrast in Table 5.

Table 7: Fixed-grid breadth and target-mode NLL decomposition for the two-worker experiment. Modulus columns are Surplus-minus-Minimum same-style contrasts; the last two columns are response-token NLL contrasts.

Seed	Mod. 29	Mod. 30	Mod. 31	Mod. 32	Diagonal	Off-diagonal
20260827	.4727	.4668	.4961	.4746	$-.005640$	$+.004235$
20260828	.4707	.4727	.4766	.4629	$-.005681$	$+.004303$
20260829	.4609	.4902	.4746	.4727	$-.005690$	$+.004383$

Across seeds, the average same-style contrast is positive on all 32 frozen specifications (range $+0.2969$ to $+0.5938$; mean $+0.4743$). Within each seed, all 32 specification-level contrasts are also positive; the three per-seed minimum contrasts are 0.3125, 0.296875, and 0.28125.

Effective support of the student style-pair law. For each seed, arm, and held-out specification, the 64 program-valid free generations define an empirical distribution over the 16 legal style pairs. We summarize it by 2^H, the number of equiprobable pairings with the same plug-in joint entropy, and by the number of distinct pairs observed. Mean effective support is 13.88, 13.76, and 14.06 for Minimum and 8.56, 8.60, and 8.42 for Surplus; paired Surplus-minus-Minimum differences are -5.32 $[-5.77, -4.86]$, -5.16 $[-5.59, -4.70]$, and -5.64 $[-6.09, -5.17]$. The corresponding observed-pair differences are -2.50 $[-3.09, -1.94]$, -2.34 $[-2.81, -1.88]$, and -2.75 $[-3.25, -2.25]$. Intervals use 10,000 paired percentile-bootstrap replicates over specifications with seed 20260826. This is a descriptive translation of the same frozen output counts, not an independent utility endpoint.

C.7 Dose-Ladder Construction and Audit

The two interior laws preserve the exact prompts, canonical responses, and 32-row budget per prompt and orientation of the two-worker design, changing only the 4×4 style-pair count matrix. The $d{=}3$ law assigns count three to each diagonal cell and a permutation of $(2, 2, 1)$ to the three off-diagonal cells of each row; the $d{=}4$ law assigns four and a permutation of $(2, 1, 1)$. The permutations keep every column sum at eight. Each worker therefore still uses every style eight times per orientation, so the 1,344 prompt–orientation–worker groups retain the endpoint laws' local multisets of strings,

ASTs, and token IDs; row counts, prompt–response support, and source and supervised tokens are unchanged.

The three off-diagonal rotations are assigned by the frozen training modulus so that each occurs seven times within every family-by-offset stratum of the 336 training specifications. Both interior laws use the same rotation for a given specification, and no arm, level, or rotation identifier appears in the prompt or response. The realized files were independently checked for the stated masses, total correlations, and balance properties.

The two interior laws were trained at paired endpoint seeds 20260827 and 20260828; endpoint students were reused rather than retrained. All four new runs completed 672 updates, and all four evaluations returned the frozen 3,584-row package: 1,024 teacher-forced uniform-target responses, 2,048 unconstrained generations, and 512 explicit controls. Program validity was 100% in every new free-generation cell. Slopes use ordinary least squares on training same-style mass; their 95% intervals come from 10,000 bootstrap replicates resampling the 32 held-out semantic specifications with the two observed trained seed pairs fixed.

Table 8: Discrete response to the exact-prompt training law. Entries are held-out same-style probabilities; endpoint students are reused from Figure 2.

Training law	Mass	TC (bits)	20260827	20260828	Pooled
Minimum (d=2)	.250	0	.2427	.2495	.2461
d=3	.375	.094	.4058	.4038	.4048
d=4	.500	.250	.5615	.5669	.5642
Surplus (d=5)	.625	.451	.7202	.7202	.7202

Regressing held-out same-style probability on training same-style mass gives slope 1.265 with a specification-bootstrap 95% interval of $[1.208, 1.321]$, conditional on the two trained seed pairs. Subtracting each student's own marginal-implied same-style probability yields pooled excesses of -0.011, $+0.149$, $+0.306$, and $+0.459$; the corresponding slope is 1.253 $[1.195, 1.309]$. The fitted slopes summarize these four discrete laws and are not a transmission law from training TC to student dependence.

C.8 Second-Relation Protocol and Confirmatory Results

The four-component intervention uses the same frozen candidate pool described in Appendix B. Candidate ordering by AST mode and source digest is a treatment construction, not an observation about how `qwen3.8-max` couples generations. Table 9 summarizes the matched data-level quantities and assigned multiplicity difference; update-level denominator equality was not audited.

Table 9: Balance of the second-relation intervention, reported per arm unless otherwise noted. Same-rank tuples place all components at one position in the frozen ordering.

Quantity	Minimum	Surplus
Exact training wordings	5 shared, byte-identical	
Legal responses/wording	2,048	
Rows/wording; rows/arm	4,096; 20,480	
Distinct prompt–responses	10,240	
Interface-state law	uniform over all 8 states	
Candidate marginals/tokens	identical per wording and state	
Source / response tokens	4,186,112 / 2,132,480	
Copies per tuple/state	2	1, plus 64 for four same-rank tuples
Candidate-identity TC	0	2.058 bits
AST-mode TC	0	1.166 bits

Training uses the two-worker experiment's response-only Qwen3-8B recipe with one epoch and 640 updates. Evaluation crosses eight byte-disjoint prompt wordings with 256 temperature-0.8 continuations each, thinking disabled and a 512-token completion limit. The executable evaluator requires exactly the four functions, runs the full pipeline, checks all three typed interfaces, and

classifies each component's AST mode. The primary endpoint subtracts the all-four-same-style rate implied by each student's own component marginals before comparing arms.

For seeds 20260827, 20260828, and 20260829, the Surplus-minus-Minimum endpoints are respectively $+0.242460$, $+0.240080$, and $+0.248870$; their 10,000-replicate paired nested-bootstrap 95% intervals are $[0.217190, 0.267300]$, $[0.215326, 0.264966]$, and $[0.220887, 0.276901]$. The pre-specified unweighted mean is $+0.243803$, with a hierarchical-bootstrap interval of $[0.227605, 0.259577]$. Every one of the 24 seed–wording contrasts is positive, and all 12,288 requested programs are valid.

C.9 Bounded Cross-Family Exact-Prompt Check

The cross-family check repeats the two-worker intervention with `meta-llama/Llama-3.1-8B-Instruct` at revision `0e9e39f249a16976918f6564b8830bc894c89659`. The single paired model–data seed, 20260827, was fixed before either outcome was read. Each arm reuses the 21,504 training rows and 10,752 distinct exact prompt–response pairs of the Qwen experiment. Under every exact prompt, both arms retain the complete 32-mode constructed legal support and identical worker-local conditional label laws. Epoch-total source and assistant token counts, optimizer settings, the 672-update budget, and checkpoint selection are matched between arms. Minimum repeats every style pair twice per orientation; Surplus repeats diagonal pairs five times and off-diagonal pairs once, changing the data-level task-free total correlation from 0 to 0.4512050593 bits. This does not establish equal optimizer-effective token weights for the Llama tokenizer.

Evaluation reuses the frozen evaluation package of the Qwen two-worker experiment: 1,024 teacher-forced uniform-legal responses, 2,048 unconstrained generations, and 512 explicit legal-style controls. The held-out semantic specification ($n = 32$) is the analysis unit, and intervals use 10,000 paired-bootstrap replicates over specifications. Invalid generations remain in all-generation denominators. Table 10 reports the frozen readouts.

Table 10: One precommitted Llama-3.1-8B-Instruct pair under the two-worker intervention. Intervals resample the 32 frozen held-out specifications and therefore quantify variation across this evaluation grid, not across training seeds or model families.

Readout	Minimum	Surplus	Surplus − Minimum
Same-style rate	.26270	.69922	+.43652 [.39453, .47119]
Own-marginal excess	.00994	.46868	+.45874 [.43489, .48355]
Uniform-target NLL	.02497	.02753	+.002558 [.002181, .003144]
Program validity	1.00000	.96191	−.03809 [−.09424, .00000]
Diagonal control adherence	.25000	.47656	+.22656
Off-diagonal control adherence	.00000	.00000	.00000

The raw same-style contrast is positive on 31 of the 32 held-out specifications and in all four held-out modulus blocks; the only negative specification has contrast -0.03125. The NLL and own-marginal-excess contrasts are positive on all 32. Restricting to program-valid generations yields a same-style contrast of $+0.46836$ $[0.44568, 0.49156]$, again positive on all 32; the localized Surplus validity difference therefore suppresses rather than creates the registered all-generation association. The explicit-control difference-in-differences is $+0.22656$ $[0.18750, 0.25000]$, but it comes entirely from stronger diagonal response bias in Surplus because off-diagonal adherence is zero in both arms.

C.10 Finite and Forest Solver Checks

In the analytic shared/local construction, generic IPF and the closed-form Minimum agree to maximum absolute probability error 2.08×10^{-17}, with marginal error at floating-point precision. The measured local-rate estimator variances, relative to i.i.d. sampling, are $0.999\times$ for Independent, $0.991\times$ for Minimum, and $8.152\times$ for Global; their Global-over-Minimum ratio of 8.23 is the value reported in Table 1. The forest implementation solves each allowed edge law by two-marginal IPF and then composes the tree-Markov product from Appendix A. On deterministic forests of 8, 64, 1,024, and 10,000 nodes, measured marginal error is zero; the corresponding single-machine solve times are 0.00011, 0.00073, 0.00981, and 0.09057 seconds (Table 11). These times are engineering

observations, not portable performance claims. The scientific claim is linear storage and ancestral sampling, plus a linear number of edge-fitting subproblems, for fixed local support on forests. End-to-end fitting time is linear in these instances because their local alphabets and per-edge solve costs are uniformly bounded.

Table 11: Exact forest factorization at increasing scale. Each node has one relation-constrained bit and one task-free bit; every edge constrains only the former, so Minimum pays one bit per edge. The Global comparator shares both coordinates across all nodes of the entire forest instance, including across different connected components, and therefore pays $2(n-1)$ bits. Marginal errors are zero at all sizes; solve times are single-machine engineering observations, not the basis of the complexity claim.

Nodes	Edges	Global TC	Minimum TC	Surcharge	Time
8	7	14	7	7	1.1×10^{-4} s
64	56	126	56	70	7.3×10^{-4} s
1,024	896	2,046	896	1,150	9.8×10^{-3} s
10,000	8,750	19,998	8,750	11,248	9.1×10^{-2} s

D Additional Experimental Details

D.1 Fixed-Denominator Refinement Boundary

Each arm of the specification-refinement experiment contains $21{,}504$ rows and $2{,}396{,}672$ response-label tokens, arranged as 672 canonical 32-row updates. Minimum repeats every exact worker-order pair twice, whereas Coupled repeats same-order pairs three times and cross-order pairs once. Intervals use $10{,}000$ paired cluster-bootstrap draws over the 128 specifications and retain every parse and execution failure in the denominator.

Writing A/D for ascending/descending worker order, Coupled emits only same-order outputs (384 DD and 128 AA). Minimum's AA/AD/DA/DD refined-success counts are $0/18/3/82$, which yields the registered -23.83-point interaction. Training contains only the coarse task and no revision demonstrations, while each initial worker order is fixed opposite its target. A separately frozen random-draw readout is measurement-ineligible and does not alter the decision.

A post-hoc four-specification diagnostic fully crosses initial and requested orders. Conditional on the initial pair, all 64 Coupled outputs are invariant to the requested target, and Minimum is invariant in 13 of 16 fixed-initial groups. Requested orientation survives only 10/64 Minimum and 7/64 Coupled outputs. This narrows the observed failure to target ignoring and partition-assignment failure, but it was motivated by the failed interaction and identifies neither an internal mechanism nor a revised endpoint.

D.2 Permutation-Null Calibration of Student-Output MI

For empirical style-pair frequencies $\hat{p}(a,b)$ and worker marginals $\hat{p}_1, \hat{p}_2$, the plug-in cross-worker mutual information and own-marginal excess are

$$I(Z_1;Z_2) = \sum_{a,b} \hat{p}(a,b) \log_2 \frac{\hat{p}(a,b)}{\hat{p}_1(a)\hat{p}_2(b)}, \qquad E_{\text{own}} = \Pr(Z_1 = Z_2) - \sum_s \hat{p}_1(s)\hat{p}_2(s).$$

For two workers, the first is the total correlation of the empirical student-output law; the second is zero under independence for arbitrary worker marginals. We define $I_{\text{above null}} = I_{\text{plugin}} - \mathbb{E}_{\text{perm}}[I_{\text{plugin}}]$.

For each seed, arm, and held-out specification, the calibration operates on the same 64 program-valid free generations used in Section 4.2. Worker 0's style sequence is held fixed and worker 1's is uniformly permuted; this preserves both empirical marginals exactly and removes only their pairing. We average plug-in MI over $4{,}096$ permutations with fixed permutation seeds. The Monte Carlo standard error of every cell-level null mean is below 0.00087 bit. Arm contrasts are computed within each specification and bootstrapped over the 32 specifications with $10{,}000$ percentile replicates using bootstrap seed 20260826. Table 13 reports the resulting cell-level estimates.

Table 12: Exact-atom carriers built from released programs. Contrasts are source-dissimilar minus seeded-uniform for CodeContests and Cross-heavy minus Within-heavy for multi-source, with paired task-cluster 95% intervals. Composition endpoints show whether regrouping reaches student outputs; registered correctness endpoints ask separately whether that change is valuable.

Carrier	student composition	Registered outcome
CodeContests, 78 survivors	fewer source/AST repeats under source-dissimilar grouping	two-attempt any-success: $3.85\% \rightarrow 2.56\%$; -1.28 points $[-4.27, +1.71]$
Multi-source, 32 development tasks	same-style among classifiable responses: $81.6\% \rightarrow 32.0\%$	any-correct: $88/256 \rightarrow 80/256$; -3.13 points $[-9.87, +3.62]$

Table 13: Student-output MI calibration by seed. Each cell reports Minimum / Surplus in bits. Above-null values subtract the fixed-marginal permutation mean; slightly negative Minimum point estimates are compatible with zero.

Quantity	20260827	20260828	20260829
Plug-in MI	.1072/.8317	.1036/.8182	.1072/.8431
Permutation-null MI	.1129/.1121	.1130/.1124	.1121/.1124
MI above null	−.0057/.7196	−.0093/.7058	−.0050/.7307

D.3 Conditioning on Generated Task-Required State

This attribution audit reuses the frozen generations from the two-worker and four-component Qwen3-8B experiments and adds neither training nor generation. For the two-worker experiment, the required state is the generated routing orientation. Within each held-out specification and orientation, we recompute same-style excess relative to the two orientation-conditional worker marginals and calibrate conditional total correlation by permuting worker styles independently within orientation. For the four-component experiment, the required state is the joint value of the three generated interface protocols. Within each held-out wording and interface state, we recompute the all-four-same excess relative to the four state-conditional component marginals and calibrate conditional total correlation by component-wise permutations within state. All contrasts remain paired within each evaluation unit (Table 14).

Table 14: Required-state-conditioned excess-style contrasts from frozen readouts. Raw is the corresponding unconditional own-marginal-adjusted Surplus-minus-Minimum contrast; conditioned recomputes the contrast within the generated task-required state.

Experiment	Seed	Raw	Conditioned	Attenuation
Two-worker	20260827	.47130	.46134	.00996
Two-worker	20260828	.46842	.45972	.00870
Two-worker	20260829	.46534	.45708	.00826
Four-component	20260827	.24246	.23824	.00422
Four-component	20260828	.24008	.23644	.00364
Four-component	20260829	.24887	.24299	.00588

For the two-worker experiment, the seed–specification cluster-bootstrap mean conditioned gap is 0.45938 with a 95% interval of $[0.445996, 0.472305]$; the calibrated conditional total-correlation gap is 0.68696 bits ($[0.650534, 0.721952]$). Both contrasts are positive on all 32 held-out specifications in every seed. For the four-component experiment, the seed–wording cluster-bootstrap mean conditioned gap is 0.23922 ($[0.228197, 0.251179]$), and the calibrated conditional total-correlation gap is 1.44014 bits ($[1.382792, 1.494779]$); both are positive on all eight held-out wordings in every seed. The two total-correlation values are not numerically comparable because the two experiments use different joint spaces. These analyses exclude required-state mixture as an explanation within the two designs.

D.4 CODECONTESTS FIXED-ATOM PROTOCOL AND FAILURE ACCOUNTING

The pilot uses `deepmind/code_contests` revision `802411c3010cb00d1b05bad57ca77365a3c699d6`. A whole-contest split keeps held-out tasks outside SFT; result-blind exact-output qualification retains 104 tasks. The 1,404 training tasks supply twelve frozen Python programs each, giving 16,848 response rows per grouped arm. A uses seeded-uniform legal matching; B maximizes source-feature distance under the same token-length-rank footrule. Program occurrence, slot, prompt, rows, supervised-token total, updates, fixed loss denominator, model, and seed match. C splits the same atoms into single-program responses and is a practical baseline. All arms fine-tune Qwen3-8B revision `b968826d9c46dd6066d109eabc6255188de91218` for 468 updates with response-only full SFT.

Each held-out task and arm produces three independent two-attempt groups from the statement and public examples; the released suite supplies the verdict. Parse failure, truncation, runtime error, timeout, and wrong answer remain in the denominator. Facility-invalid verdicts trigger whole-task, all-arm exclusion under the frozen symmetric rule, leaving 78 of 104 tasks. Of 1,404 candidate programs, 24 succeed, 939 give a wrong answer, 347 fail at runtime, 36 time out, and 58 truncate. Success counts are $10/6/8$ for random/source-dissimilar/single-solution training. The correctness floor is therefore low, and the task-bootstrap interval in Table 12 conditions on the survivors and one training seed.

A later domain audit finds invalid generated inputs in three task suites, but no verdict changes: all 36 candidates terminating in generated tests already fail a valid earlier public or private test; the four candidates whose first wrong answer occurs on a generated test share a separately confirmed valid input and reference. A post-hoc prefix check also finds a correctness-stratum mismatch: all second-slot training targets follow dataset-correct first programs, but only $11/468$ first-slot inference programs pass ($7/234$ for A, $4/234$ for B). These diagnostics limit what this low-reach carrier can show; they do not identify the cause of the utility result.

D.5 MULTI-SOURCE CARRIER QUALIFICATION AND DIRECT-UTILITY TEST

Accepted Python atoms come from the `qwen3_30b` and `deepseek` splits of `t2ance/code-solutions` revision `eff8865b42b8badf2d8f7912b0ab9cfe5df3dc1a`. Within-heavy and Cross-heavy retain the same 12,990 distinct ordered response pairs and exact atom/slot marginals while changing same-source pair mass from 72.73% to 18.18%. Training atoms precede 2025; evaluation uses the 112 post-2025 stdin-based AtCoder tasks in pinned LiveCodeBench `test6.jsonl`.

Before training, SHA-256 ordering within difficulty assigns 8 easy, 8 medium, and 16 hard tasks to development and holds back 80 tasks. A qualification check uses two code-only Qwen3-8B requests per development task (seeds 20260911/20260912, temperature 0.7, top-p 0.95, cap 2,048 tokens). Five of six readiness rules pass, but normalized-AST-distinct pairs reach only $21/32$ against a required $24/32$, so that protocol stops before training or confirmation use. This is a syntactic qualification failure, not evidence about semantic diversity.

A separate direct-utility protocol then uses the same already-open development tasks. Each arm contains 28,512 physical rows and 13,996,202 supervised tokens, with matching exact atoms, slot marginals, support, task-block order, fixed per-update denominator, optimizer, training seed 20260913, and 216 updates. The base is Qwen3-8B revision `b968826d9c46dd6066d109eabc6255188de91218`. Checkpoint-216 receives eight paired decoding seeds per task at temperature 0.7, top-p 0.95, and a 4,096-token cap. Invalid and truncated programs count as incorrect.

The registered endpoint is whether either program passes every released test: $88/256$ for Within versus $80/256$ for Cross. A paired task-cluster bootstrap retains all eight seed rows, uses 10,000 draws at seed 20260929 and finite-cluster centering correction. The Cross-minus-Within contrast is -0.03125 with 95% interval $[-0.09872, 0.03622]$, failing the frozen $+0.05$ continuation threshold. Single-program correctness changes -0.02539 $[-0.07302, 0.02025]$; response validity is $241/256$ versus $244/256$.

The both-wrong difference $+0.03125$ decomposes into -0.00781 in within-response common error and $+0.03906$ in the product of marginal errors. The common-error interval $[-0.04183, 0.01827]$ crosses zero. Composition changes nonetheless: Cross versus Within has $77/241$ versus $199/244$

same-style pairs among classifiable responses, zero versus eight exact duplicates, and one versus twelve normalized-AST matches. This development-split utility test does not consume the held-back 80 tasks. Its correctness is released-test passage, not full AtCoder judge acceptance; some Cross-only gains violate task constraints outside the released cases.

Withdrawn readouts on the 80-task split. A later companion protocol retains matched training and response provenance, but illegal inputs, invalid references, unsupported multiple-answer matching, numeric fallbacks, and incomplete validators invalidate its correctness and shared-error readouts. Those readouts are withdrawn. Normalized-AST equality remains a composition result: $110/640$ to $56/640$, difference -0.084375 $[-0.13001, -0.04035]$. The official-example and independently qualified unseen-input analyses below use separate evaluators; they do not rehabilitate that endpoint.

D.6 Error Redundancy on a Fixed Public-Example Population

An earlier Within/Cross comparison uses an independently seeded training pair and the fixed 80-task official-public-example population. These are the earlier pairing arms, not the new clone-controlled treatment of Appendix D.8. Training program inventory, per-slot marginals, and material-quality composition match. Both arms return $640/640$ responses with one provider attempt each, and all $7{,}072$ execution cells remain in the denominator. Outputs whose correctness the examples do not determine enter conservative bounds.

Native same-wrong agreement asks whether both programs in one response return an identical wrong output on the same example. Subtracting agreement between programs from different responses to that task gives an excess; the preregistered Within-minus-Cross excess interaction is conservatively $+0.065867$, with descriptive task-cluster 95% interval $[0.027994, 0.105023]$. This establishes an error-redundancy contrast on public examples, not whole-program judge acceptance.

A post-hoc decomposition separates full-denominator additive components. The same-normalized-surface component is $+0.052493$ $[0.023679, 0.083111]$; the different-surface component is $+0.013374$ $[-0.023104, 0.049964]$. The different-AST component is also unresolved ($+0.03456$, $[-0.002827, 0.072936]$). Total joint failure is not point-identified: its conservative lower interaction is -0.027102 $[-0.084636, 0.015979]$, while the unknown-output upper bound permits a positive value.

Source-linked cases illustrate both directions. On `abc395_e`, one Within pair at decoding seed 0 repeats the same wrong output on $17/24$ independently checked cases using near-duplicate templates. On `abc391_f`, a Cross pair at seed 5 repeats the same wrong output on $22/24$ cases, whereas the corresponding Within pair has zero same-wrong cases. Zero agreement need not mean correctness: distinct Cross implementations can share the same wrong mathematical target. These cases explain why syntactic redundancy and correctness must be measured separately.

D.7 A Fixed Two-of-Three Decision Rule on the Same Population

The redundancy analysis motivated an agreement-based decision rule: take both programs from one response and a third uniformly from either slot of another response to the same task; accept when two comparable outputs agree exactly, otherwise reject. Correct acceptance, wrong acceptance, and rejection partition its outcomes. Truth-unknown outputs stay in conservative bounds. Samples are equally weighted within each of the 80 tasks and tasks equally across the population, retaining all public examples, eight decoding seeds, and both slots.

The rule was selected on an earlier development pair and applied to the checkpoint pair already examined in Appendix D.6. This is retrospective consistency evidence. The subsequent comparison freezes it before two new treatment trainings with existing Cross controls (Appendix D.8), then reuses their responses on independently qualified unseen inputs (Appendix D.9).

Within coverage is 64.61% versus Cross's 57.26%, a $+7.36$-point difference with descriptive equal-task 95% interval $[+3.41, +11.50]$. Wrong acceptance changes by $+6.67$ to $+9.13$ points (lower-bound interval $[+2.69, +10.65]$), while correct acceptance changes by -1.77 to $+0.69$. Additional coverage is predominantly erroneous. A diagnostic instead draws all three votes from different responses: its direct wrong-acceptance difference is $+1.94$ to $+2.39$ points, with the lower interval crossing zero. The native-minus-repaired arm interaction is $+4.65$ to $+6.82$ points (lower interval

$[+1.25, +8.17]$). This preserves output marginals but consumes three responses rather than two, so it is an attribution check, not a cost-matched voting policy.

A post-result syntactic sensitivity allows at most one vote per consistent-renaming surface, and requires two retained surfaces to agree. It removes 13.05 points of Within acceptance (5.29 known-correct, 7.45 known-wrong, 0.31 truth-unknown) and 5.49 of Cross acceptance (1.41, 3.62, 0.47 respectively). Coverage becomes 51.56% and 51.76%; the conservative wrong-acceptance interval is $[-0.20, +6.88]$ points. Deduplication removes correct and wrong consensus, and a residual wrong-acceptance contrast is unconfirmed.

D.8 Prospective Consumer Design and Official-Example Readout

The retrospective analyses in Appendices D.6 and D.7 motivated a test that froze the two-of-three consumer, training comparison, and primary readout before the new treatment training and this comparison's program generation. Cross checkpoints and the task population had already been used in the earlier analyses. The prospective component is the new treatment comparison, not an untouched control or task replication. Appendix D.9 re-scores the resulting archived responses on independently qualified inputs absent from these examples.

Construction and qualification. Each training task uses one exact prompt and six accepted program atoms. Cross and the provenance-proxy treatment preserve per-slot program frequencies, all 30 legal ordered two-program tuples, 66 response row positions per task, and normalized exact-clone cell counts. Each arm has $28{,}512$ physical rows, equal supervised-token exposure, and 216 fixed-denominator optimizer updates. The assigned difference is the multiplicity with which complete response types occupy those positions; the teacher source label never enters the student prompt. At seeds 20260914 and 20260916, two newly trained treatment checkpoints are paired with the existing Cross checkpoints. All four Qwen3-8B runs pass the frozen terminal qualification: four workers, 216 finite updates, complete canonical-index and fixed-denominator events, no restarts, and checkpoint-216.

Evaluation and statistical unit. Each checkpoint generates eight predeclared response seeds on 80 tasks, giving $2{,}560$ archived responses and $14{,}144$ execution cells over 221 official public examples; every cell is retained. Samples are weighted equally within task and tasks equally across the population. The task is the bootstrap unit. With only two training seeds, the reported intervals are descriptive over tasks conditional on those seeds, not estimates of a training-seed distribution. Official public examples are not hidden-test utility, and this audit does not reuse the withdrawn correctness semantics of Appendix D.5.

Frozen two-of-three readout. Relative to Cross, the treatment raises the conservative wrong-acceptance lower bound by $+0.0404$, with a descriptive task-bootstrap 95% interval $[+0.0092, +0.0730]$; the upper-bound difference is $+0.0648$. The two seed-specific lower-bound point estimates are $+0.0377$ and $+0.0390$. Coverage rises $+0.0459$, while correct acceptance changes by -0.0189 to $+0.0055$ and the erroneous fraction of accepted decisions by $+0.0332$ to $+0.0736$. A diagnostic drawing the three votes from different responses leaves a native-minus-repaired conservative wrong-acceptance interaction of $+0.0396$ $[+0.0101, +0.0693]$. It consumes an extra response and is an attribution check, not an equal-cost consumer.

Three-of-four voting and candidate-set all-wrong. The frozen two-of-three rule omits one of the four programs in its two responses. Under a three-of-four rule, the pooled wrong-acceptance lower- and upper-bound differences are only $+0.0005$ and $+0.0039$, respectively, with descriptive intervals crossing zero and opposite seed-specific signs. Much of the original excess arises where an omitted vote turns a 2:2 conflict into a majority; the unseen-input enumeration measures this structure directly (Appendix D.9). The stricter rule reduces treatment coverage from 62.82% to 33.32% and correct acceptance from 32.43–34.31% to 25.89–26.28%; it is not a cost-free repair.

Five constructive problems contribute twelve official public sample cases. A static audit against their problem-specific feasibility rules resolves all 161 distinct comparable archived outputs without executing generated programs or changing the original evaluator. Of 153 previously unresolved outputs, 151 are wrong and two correct; eight already known correct remain correct. The frozen

80-task, 221-sample, two-seed denominator is unchanged. With this truth, the treatment-minus-Cross difference in the probability that all four candidate programs are wrong is $+0.04787946$ at seed 20260914 and $+0.00186012$ at seed 20260916, pooled $+0.02486979$ with a descriptive 10,000-draw equal-task bootstrap interval $[-0.00333054, +0.05268043]$. This candidate-set all-wrong readout is post-hoc, distinct from the frozen two-of-three endpoint; its pooled interval crosses zero.

Own-inventory rematching reference. Within each task, sample, seed and arm, hold the eight first-slot and eight second-slot outcomes fixed and average over uniform reassignments of the second-slot positions. Exact enumeration agrees with the closed-form reference in all 884 groups. Writing F for the observed candidate-set all-wrong rate, F_0 for that arm's own-inventory reference and Δ for the treatment-minus-Cross difference gives

$$\Delta F = \Delta F_0 + \Delta(F - F_0).$$

The inventory terms ΔF_0 are $+0.03541002$ and -0.01013233 in the two seeds; the pairing-excess differences $\Delta(F - F_0)$ are $+0.01246944$ and $+0.01199245$, pooled $+0.01223$ with a descriptive equal-task interval $[+0.00413, +0.02061]$. Comparable programs known to be wrong, rather than only invalid or missing outputs, carry the positive pooled pairing excess. The negative inventory term nearly cancels the second seed's all-wrong total. This accounting is not a randomized training null, identified mediator, pure dependence effect or deployable rematching policy: each reference conditions on its arm's own generated slot marginals.

Exact normalized clones are balanced, but broader token- and strategy-level similarity is not. The experiment therefore identifies the assembled complete-response law, rather than a separate source-label effect.

D.9 Unseen-Input Confirmation under the Frozen Consumer

The prospective test above scores the frozen consumer on official public examples. To check whether its decision effect is peculiar to that displayed set, we reuse the four qualified checkpoints—control and treatment at seeds 20260914 and 20260916—their archived eight two-program responses per task and arm, and the original consumer unchanged. This analysis adds no student training, program generation, model call, or consumer modification.

Input qualification and statistical unit. Source-semantic qualification, completed before new program results were examined, retained all 75 deterministic-output tasks from the frozen 80-task family. Five constructive-output tasks were excluded because one reference output cannot fairly define exact correctness for them. The qualified set contains 1,186 distinct official-domain inputs absent from the public examples: 16 per task except `abc393_a`, whose complete official input domain has only two unseen members. Program stdout was assessed in the actual qualified Python 3.9.6 sandbox, with parse and execution failures retained in the fixed denominator of 75,904 logical cells. Per-task mean rates are averaged equally across tasks and the two fixed training seeds; the paired task bootstrap is descriptive conditional on those seeds, tasks, inputs, and sandbox.

Frozen two-of-three readout. Wrong acceptance is 23.1265% under control and 27.4382% under treatment, a difference of $+4.3118$ percentage points with a paired task-bootstrap 95% interval of $[+1.4631, +7.3372]$. The seed-specific differences are $+3.8170$ and $+4.8065$ points, satisfying the frozen materiality rule of at least $+3$ points, a positive task-bootstrap lower bound, and a positive contrast at both seeds. Correct acceptance is 38.9397% versus 39.6049% ($+0.6652$ points), while total acceptance is 62.0662% versus 67.0432% ($+4.9769$ points). Most additional accepted decisions are wrong, though correct acceptance does not fall. Task-level wrong-acceptance effects are positive on 43 tasks, negative on 25, and zero on 7; low- and high-small-input strata give $+3.9047$ and $+4.7870$ points, with one task in a single stratum.

Voting-rule and candidate-correctness boundaries. Requiring at least three of all four candidate outputs to agree gives a wrong-acceptance difference of $+0.1756$ points, with descriptive interval $[-1.0259, +1.3726]$. A rejection under this rule means that no output receives three votes; it does not mean that all four candidate programs are wrong. A separately versioned read-only audit gives the actual candidate-set all-wrong difference as $+1.2723$ points, interval $[-1.9951, +4.2941]$.

Vote-structure decomposition. A post-hoc read-only enumeration of the same archive forms every unordered pair of eight two-program responses in each task–input–seed–arm cell and all four ways of omitting one slot. Invalid and noncomparable slots remain non-votes; tasks and the two fixed seeds retain the frozen weights. The 132,832 four-slot units and 531,328 three-slot decisions reproduce both voting-rule point estimates. For one four-slot unit, let R_3^{wrong} be the fraction of its four triples that accept wrongly and R_4^{wrong} indicate a wrong output receiving at least three of the four votes. Direct enumeration and an independent count check agree in every unit that

$$R_3^{\mathrm{wrong}} = R_4^{\mathrm{wrong}} + \tfrac{1}{2}\#\{\text{distinct wrong outputs occurring exactly twice in four slots}\}.$$

Under the frozen weighting, the $+4.3118$-point treatment-minus-Cross contrast partitions into $+0.1756$ points already admitted by four votes, $+5.1563$ from two wrong votes within one response, and -1.0201 from two wrong votes split across responses. The within-response term is positive in both seeds ($+4.2664$ and $+6.0461$ points). An alternative grouping of the same total assigns $+4.0811$ points to complete four-comparable 2:2 conflicts and $+0.0551$ to other exactly-two-wrong-output configurations; these groupings are not additive. Within the 2:2 contribution, $+3.0060$ points pit two different wrong answers against each other and $+1.0751$ pit a wrong answer against a correct one. Because 54,919 units have fewer than four comparable slots, the residual configurations cannot all be called four-valid 2:1:1 splits. The counts explain the frozen consumer's rule sensitivity, not a training mediator; its primary interval and decision are unchanged.

Two post-hoc cases illustrate sign heterogeneity, not a mechanism. On `abc388_d`, input $N = 4$ with values $(0, 5, 8, 7)$ has true output $(0, 3, 8, 9)$. At seed 20260914 neither arm had a correct candidate on that input, while treatment repeated wrong outputs such as $(0, 5, 13, 20)$; the task-level wrong-acceptance contrast across its new inputs is $+50.06$ points. An archived treatment response pair for this task and seed contains $(0, 5, 13, 20)$ in both slots of one response and a different wrong output, $(0, 6, 10, 10)$, in both slots of the other; every three-slot reading accepts wrongly while the four-slot rule rejects. On `abc400_e`, query 61 has true output 36; the same seed's control had no correct candidate and treatment had four, while the task-level contrast is -23.83 points. An archived treatment pair for this task and seed has three correct votes for 36 and one wrong vote, so both rules accept correctly. Candidate quality can offset the dependence effect.

This extends the official-example finding to qualified unseen inputs, reusing the same checkpoints, task family, and consumer rather than providing an independent training or benchmark replication.

D.10 Fixed-First-Prefix Continuation and Bounded Missingness

Table 15: Fixed-first-program continuation. Upper block: one unseen input; lower block: whole-suite program success. Lower-block percentages are shares of *all* paired attempts, not conditional rates among first failures. Ranges bound one unknown continuation.

Measure	Cross	Treatment	Treatment − Cross
Per unseen input: same-wrong matching			
Different-prefix baseline	5.872–5.917%	5.704%	−0.213–−0.168 pt
Indexed excess D	14.364–14.453%	20.042%	+5.590–+5.679 pt
Per program: passes every unseen input of its task			
Second	22.044–22.088%	21.972%	−0.116–−0.072 pt
Both	18.430%	19.310%	+0.881 pt
Second rescues a failed first	3.614–3.658%	2.662%	−0.997–−0.952 pt

This continuation test reuses the four qualified checkpoints of Appendix D.8 and the 75 deterministic-output tasks of Appendix D.9, but unlike the archived-vote analysis it generates new second programs. Its frozen native-boundary admission keeps 2,324 of 2,400 archived first-program prefixes, with both arms and seeds on every task. Each admitted prefix is continued once by each same-seed checkpoint with the same request and decoder. The first program is therefore identical *within this crossover*; the test does not hold the first-program distribution fixed between the separate archived-voter arms above. The frozen unseen-input suite yields 73,500 logical task–input–prefix–arm cells. Of 4,648 formal continuations, 4,647 were observed and 73,484 cells measured; one continuation and its 16 cells are permanently unknown, not inferred from timeout or silently excluded.

For each task/input/seed/checkpoint cell with at least ten admitted prefixes, same-wrong equality requires both programs to produce comparable executable outputs, both to be wrong, and their canonical output keys to match. The indexed-pair rate is compared with the mean over all ordered distinct-prefix pairs in that cell. Input rates are averaged within tasks and the two fixed seeds, then tasks weighted equally. The sole primary contrast is treatment-minus-Cross indexed excess D. Cross's D is 14.364–14.453% and treatment's is 20.042%; the difference is $+5.590$–$+5.679$ points. Seed-specific bounds are $+4.600$–$+4.778$ and $+6.579$ points. The paired $10{,}000$-draw task-bootstrap outer 95% interval is $[+4.001, +7.310]$ points, conditional on the fixed tasks, seeds, checkpoints, inputs and sandbox. Universal assignments of the missing second output give all reported ranges.

Whole-suite correctness requires passing *every* frozen input for the task (Table 15). The shared first program is all-correct on exactly 21.354% of attempts in both arms. At-least-one success is 24.967–25.012% for Cross versus 24.015% for treatment, a treatment-minus-Cross difference of -0.997 to -0.952 point, with paired-task bootstrap outer 95% $[-1.822, -0.213]$. The identity $P(\text{at least one all-correct}) = P(\text{first all-correct}) + P(\text{first not all-correct, second all-correct})$ explains this whole-suite difference; the input-level same-wrong cells do not decompose it. Different-prefix whole-suite any-correct changes by approximately zero, with an interval spanning zero.

These rescue percentages are joint shares of all paired task-prefix attempts, not success rates conditional on first-program failure. Relative to Cross's incremental rescue mass, treatment forgoes 26.35–27.25% of rescues; overall at-least-one success falls by less than one percentage point. Both seed-specific rescue contrasts are negative even though one seed's marginal second-program all-correct rate rises 0.167–0.256 point. The pooled second-program marginal difference is -0.116 to -0.072 point and its uncertainty spans zero, so the data do not establish equivalence of the second programs' standalone quality.

Later local measurement was scheduled at lower priority without changing requests, cases, or per-case limits. Seventy-eight genuine wall-clock timeouts were observed in that later segment. Different programs were measured before and after the scheduling change, so the timeout cause is not identified. Even allowing all their second-program outputs to change adversarially leaves a pooled primary lower bound of $+5.176$ points and bootstrap lower endpoint $+3.522$ points. At most two whole-suite program statuses could become all-correct, insufficient to reverse the negative pooled any-correct point estimate; exact magnitudes remain environment-conditional. On `abc388_b` at seed 20260914, the same wrong first prefix using $T + kL$ is followed by a Cross continuation using the correct $T(L + k)$ relation on all 16 unseen inputs, while treatment returns the first program's wrong outputs on all 16. The opposite direction exists: a seed-20260916 prefix on the same task has a Cross continuation that fails at runtime on all 16 inputs and a treatment continuation correct on all 16. These are illustrations, not an identified copying mechanism or universally helpful pairing rule.

D.11 Training and Evaluation Configuration

The exact-prompt experiment uses Qwen3-8B revision `b968826d9c46dd6066d109eabc6255188de91218`, the native Qwen3 non-thinking chat template, and response-only full-parameter SFT. We train with AdamW (betas $0.9/0.95$, weight decay 0.1), learning rate 2×10^{-5}, a cosine schedule with 10 warmup steps, global batch 32 on four GPUs (per-device batch 1, gradient accumulation 8), BF16, gradient checkpointing, FlashAttention-2, and ZeRO-3. Each exact-prompt run uses one epoch and 672 optimizer updates; only checkpoint-672 is read. The paired seeds are 20260827–20260829. Appendix C.2 gives the original per-window loss normalization and its source-stack sampler replay. The matched-denominator rerun keeps this configuration at paired seed 20260827 while replacing the realized per-window count with one fixed supervised-token denominator common to both arms. The validation-law selection audit then recomputes complete-response NLL on the frozen held-out rows without retraining or filtering them (Appendices C.3 and C.4). The graded intervention reuses this configuration for its two interior laws at seeds 20260827 and 20260828; the endpoint students are the runs above, reused rather than retrained (Appendix C.7). The second-family repetition uses `meta-llama/Llama-3.1-8B-Instruct` at revision `0e9e39f249a16976918f6564b8830bc894c89659` with the single paired seed 20260827, fixed before outcomes; the training and evaluation configuration otherwise matches the two-worker experiment (Appendix C.9).

For free generation, each held-out specification receives 64 continuations with thinking disabled, temperature 0.8, top-p 0.95, top-k 50, at most 384 new tokens, and one continuation per row. The

executable oracle and AST classifier determine validity and the two worker styles. Explicit-control evaluation crosses every held-out specification with all 16 requested style pairs under greedy, thinking-disabled decoding; a request counts as adhered to only when both the program oracle and both AST classifications pass. Every requested generation is retained, and invalid generations remain in the denominator.

For the two-worker experiment, uncertainty intervals use $10{,}000$ paired bootstrap replicates over the 32 semantic specifications. The analysis unit is a held-out specification rather than a token or continuation, and the three training seeds are reported separately as directional replications rather than pooled into a population interval. The four-component experiment uses paired nested bootstrap intervals within each seed and a separately pre-specified hierarchical summary that resamples its three fixed training seeds, held-out wordings, and generations.